JACQUES BOISGONTIER



ISBN : 2.86595.120.X

52 programmes

ORIC-1 pour tous

INITIATION GENERALE

Visa pour l'informatique — Jean-Michel Jégo
Mon ordinateur — Jean-Claude Barbance
L'ordinateur individuel — Yves Leclerc

INITIATION AUX MATERIELS :

Collection « ... *POUR TOUS* »

ORIC-1 pour tous — Jacques Boisgontier et Sophie Brébion

Collection « *MATERIELS* » — *Série verte*

La découverte de l'Apple II — tome 1 — par Frédéric Lévy et Dominique Schraen
La découverte du CBM — par Daniel-Jean David
La découverte du VIC — par Daniel-Jean David
La découverte du PC-1211 — par Jean-Pierre Richard
La découverte de la TI 57 — par Xavier de la Tullaye
La découverte du PC-1251 — par Jean-Pierre Richard
La découverte du PC-1500 — par Jean-Pierre Richard
La découverte du FX 702P — par Jean-Pierre Richard
La découverte du TI-99/4A — par Frédéric Lévy et Dominique Schraen
La découverte de l'Atari — par Daniel-Jean David
La découverte du PB-100 — par Pierrick Moigneau
La découverte du Goupil — par Jean-Yves Michel
La découverte du TO7 — par Dominique Schraen et Maurice Charbit

INITIATION AUX LANGAGES

Collection «... POUR TOUS »

BASIC pour tous — Jacques Boisgontier et Sophie Brébion

Collection « LANGAGES » série verte

Langages de programmation — Stéphane Berche et Claude Lhermitte
Programmer en Basic — Michel Plouin
Programmer en LSE — Stéphane Berche et Yves Noyelle

Autres ouvrages relatifs à l'ORIC-1 :

ORIC-1 pour tous — Jacques Boisgontier et Sophie Brébion
Collection « POUR TOUS »

La découverte de l'ORIC — Daniel-Jean David
Collection « MATERIELS » — Série verte

ISBN : 2-86595-120-X

52 programmes ORIC-1 pour tous

par
Jacques Boisgontier



ÉDITIONS DU P.S.I.
1984

SOMMAIRE

Première partie
EXERCICES

EDITIONS

Nous approfondissons dans ce chapitre quelques instructions de l'ORIC-1.

PRINT TAB() :

PRINT TAB() présente deux anomalies :
Il faut ajouter 13 pour obtenir la tabulation souhaitée.

Exemple :

```
10 PRINT TAB(13+5);"ORIC"
```

Affiche 'ORIC' en colonne 5.

```
10 PRINT "DUPONT";TAB(13+8);"JEAN"
```

N'affiche pas 'JEAN' en colonne 8 comme cela devrait être le cas.
On procède donc ainsi :

```
10 NOM$="DUPONT":PR$="JEAN"
20 PRINT NOM$;SPC(20-LEN(NOM$));
30 PRINT PR$
RUN
DUPONT              JEAN
|<------ 20 ------>|
```

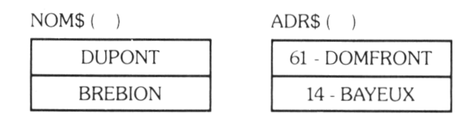
LPRINT TAB() :

Pour obtenir la tabulation verticale, il faut ajouter 13 :

```
10 NOM$="DUPONT":PR$="JEAN"
20 LPRINT NOM$;TAB(13+20);PR$
```

Nous présentons ci-dessous un programme d'édition d'étiquettes. Les adresses sont stockées en DATA.

Plusieurs étiquettes sont imprimées sur une ligne. Par conséquent, il faut utiliser des tables **'tampons'** pour stocker puis imprimer les noms et adresses.



Etiquettes

```
10 REM        ETIQUETTES
20 :
30 IH=22              ' Intervalle entre etiquettes
40 NN=6               ' Nombre d'etiquettes
50 :
60 DATA MR DUPONT,    61-DOMFRONT
70 DATA MR BREBION,   14-BAYEUX
80 DATA MR BALUTIN,   75-PARIS
90 DATA MR ROULET,    75-PARIS
100 DATA ME MARION,    77-LAGNY
110 DATA MR DEGREMONT,92-ST CLOUD
120 :
140 FOR K=1 TO INT(NN/3)
150 :FOR I=1 TO 3
160 : READ NOM$(I),ADR$(I)
170 :NEXT I
180 :
200 :FOR I=1 TO 3
210 :  LPRINT TAB(13+(I-1)*IH);
220 :  LPRINT NOM$(I);
230 :NEXT I
240 :LPRINT
250 :
260 :FOR I=1 TO 3
270 :  LPRINT TAB(13+(I-1)*IH+2);
280 :  LPRINT ADR$(I);
290 :NEXT I
300 :LPRINT:LPRINT
310 NEXT K
```

```
MR DUPONT              MR BREBION             MR BALUTIN
  61-DOMFRONT            14-BAYEUX              75-PARIS

MR ROULET              ME MARION              MR DEGREMONT
  75-PARIS               77-LAGNY               92-ST CLOUD
```

Editions imprimantes :

Pour éviter que l'imprimante 'oublie' des caractères, on fait :

```
125 CALL #ED01          ' Clavier inhibe
126 LPRINT "COUCOU"
127 CALL #ECC7          ' Activation clavier
```

Pour lister un programme, on fait :

```
 2 CALL #ED01:LLIST 10-
10 Programme a lister
```

Couleurs avec PLOT X, Y :

Avec PLOT, les couleurs d'écriture et de fond peuvent être déterminées en ajoutant à la chaîne à imprimer

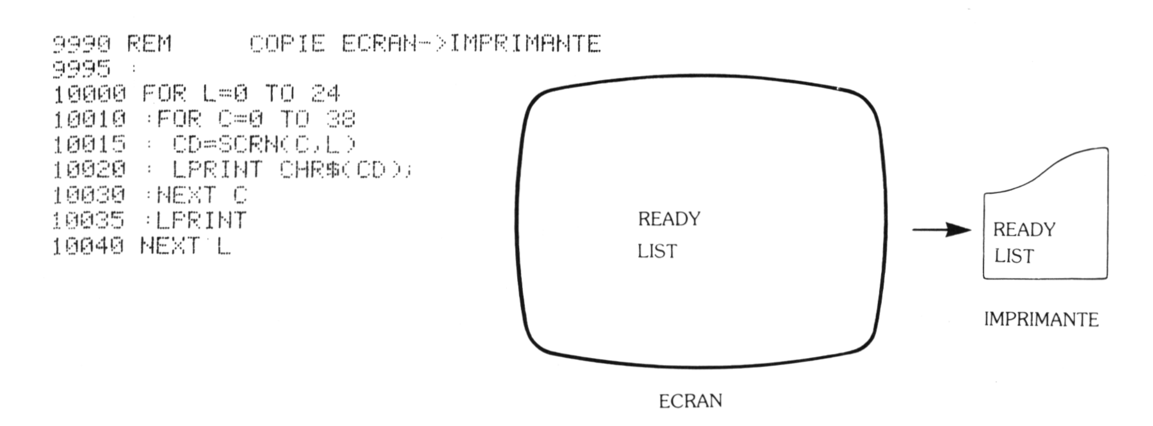
- **CHR$** (couleur écriture)
- **CHR$** (16 + couleur fond)

```
10 CLS
20 :
30 X$=CHR$(2)+"VERT"
40 PLOT 10,10,X$
50 :
60 Y$=CHR$(16+1)+"FOND ROUGE"+CHR$(16+7)
70 PLOT 12,12,Y$
```

COPIE D'ECRAN

Pour recopier un écran sur imprimante, nous utilisons l'instruction **SCRN(C,L)** qui fournit le code ASCII du caractère de la position spécifiée.

Copie écran → imprimante

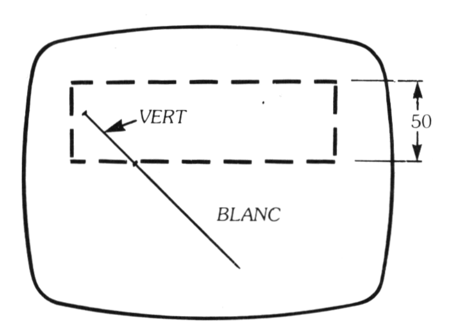


```
9990 REM      COPIE ECRAN->IMPRIMANTE
9995 :
10000 FOR L=0 TO 24
10010 :FOR C=0 TO 38
10015 : CD=SCRN(C,L)
10020 : LPRINT CHR$(CD);
10030 :NEXT C
10035 :LPRINT
10040 NEXT L
```


FILL :

Fill permet de colorier des parties d'écran.
Il permet également de modifier la couleur d'écriture (foreground)

```
10 HIRES
20 CURSET 0,0,2:FILL 50,1,2
30 :
40 CURSET 0,0,1:DRAW 100,100,1
```



KEY$

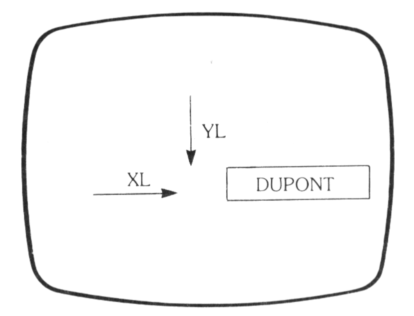
Avec la plupart des Basics, lorsque nous appuyons sur la touche <**RETURN**> en réponse à une instruction **"INPUT"**, la variable spécifiée dans l'instruction prend une valeur nulle. Sur ORIC-1, il faut **nécessairement** entrer une valeur avant d'appuyer sur <**RETURN**>, ce qui n'est pas pratique.

Nous proposons ici un sous-programme qui remplace l'instruction **'INPUT'** tout en permettant d'appuyer directement sur <**RETURN**>.
A chaque fois qu'un caractère est saisi par **'KEY$'**, nous le concaténons à 'LIG$'



Le code de suppression de caractère (en cas d'erreur de frappe) est géré par le programme (code 127 de la touche DEL).

La **première version** de programme présentée saisie une ligne là où se trouve le curseur au moment de l'appel du programme.
La **deuxième version,** saisit une ligne dans un endroit de l'écran spécifié. Nous utilisons pour cela l'instruction **PLOT X,Y.**



INPUT avec KEY$

```
2990 REM============== INPUT AVEC KEY$ ====
3000 LIG$=""
3010 :
3020 :
3030 :
3040 C$=KEY$:IF C$="" THEN 3040
3050 C=ASC(C$):L=LEN(LIG$)
3060 IF C=13 THEN PRINT:RETURN
3065 IF C<>127 THEN 3100
3070 :
3080 IF L>0 THEN LIG$=LEFT$(LIG$,L-1):PRINT CHR$(127);:GOTO 3040 EL
SE 3040
3090 :
3100 IF C<32 THEN PING:GOTO 3040
3110 LIG$=LIG$+C$:PRINT C$;
3120 GOTO 3040
```

Saisie avec KEY$ et PLOT

```
10 REM SLIG  SAISIE AVEC KEY$ ET PLOT
15 :
20 CLS
30 LIG$=""
40 XL=10:YL=10                ' Coordonnees affichage
50 :
60 L=LEN(LIG$)
70 PLOT XL+L,YL,130           ' Curseur
80 :
90 C$=KEY$:IF C$="" THEN 90 ' Attente caractere
100 C=ASC(C$)
110 IF C=13 THEN 200          ' Retour chariot
120 IF C<>127 THEN 150        ' Suppression
130 :
140 IF L>0 THEN LIG$=LEFT$(LIG$,L-1):PLOT XL+L,YL," ":GOTO 60 ELSE
90
145 :
150 IF C<32 THEN PING:GOTO 90
155 LIG$=LIG$+C$
160 PLOT XL,YL,LIG$
170 GOTO 60
180 :
200 PLOT XL+L,YL," "
210 :
```

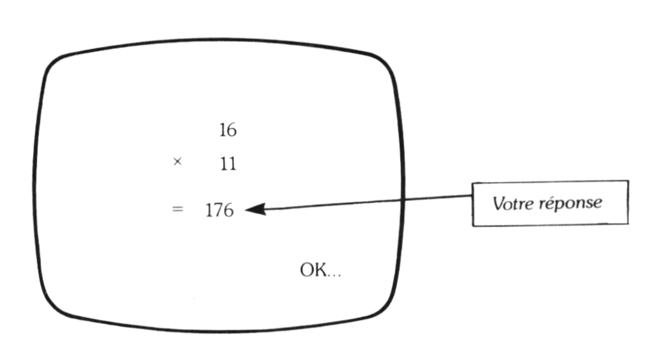
Test clavier avec PEEK (# 208) :

Le test des touches peut se faire par la lecture de la mémoire # 208.

```
10 REM  DEPLACEMENT RAPIDE
20 :
30 PRINT CHR$(6);                :REM clavier muet
40 LORES 0
50 XC=10:YC=10:PLOT XC,YC,17   :REM rouge
60 :
70 X=PEEK(#208):IF X=56 THEN 70 :REM attente
80 :
90 PLOT XC,YC,16
100 :
110 IF X=172 THEN IF XC>0 THEN XC=XC-1
120 IF X=188 THEN IF XC<37 THEN XC=XC+1
130 IF X=180 THEN IF YC<23 THEN YC=YC+1
140 IF X=156 THEN IF YC>0 THEN YC=YC-1
190 PLOT XC,YC,17
200 GOTO 70
```

Le programme ci-dessous affiche 2 nombres au hasard et demande quel est leur produit.

Multiplication




```
10 REM MULT       MULTIPLICATION
15 :
20 CLS
30 :
40 XL=10:YL=10                :REM Affichage ligne
41 REM----------- 2 nombres entre 1 et 20 ------
43 N1=INT(RND(1)*20):N2=INT(RND(1)*20)
45 IF N1<5 OR N2<5 THEN GOTO 43
46 :
50 PLOT XL-1,YL-4,STR$(N1)
52 X$="X"+STR$(N2):PLOT XL-2,YL-3,X$
53 :
55 PLOT XL-1,YL,"="
56 :
57 GOSUB 62
58 IF VAL(LIG$)=N1*N2 THEN PLOT 15,15,"OK....."
59 WAIT 200:GOTO 20
60 REM------------------ Saisie d'une ligne
62 LIG$=""
64 :
65 L=LEN(LIG$)
70 PLOT XL+L,YL,130
80 :
90 C$=KEY$:IF C$="" THEN 90
100 C=ASC(C$)
110 IF C=13 THEN 200
120 IF C<>127 THEN 150      :REM Suppression
130 :
140 IF L>0 THEN LIG$=LEFT$(LIG$,L-1):PLOT XL+L,YL," ":GOTO 65 ELSE
90
145 :
150 IF C<32 THEN PING:GOTO 90
155 LIG$=LIG$+C$
160 PLOT XL,YL,LIG$
170 GOTO 65
180 :
200 PLOT XL+L,YL," "
210 RETURN
```

CARACTÈRES SPÉCIAUX

Les caractères spéciaux permettent d'afficher des messages de couleurs différentes, clignotants, en double hauteur. Nous présentons ici quelques exemples.

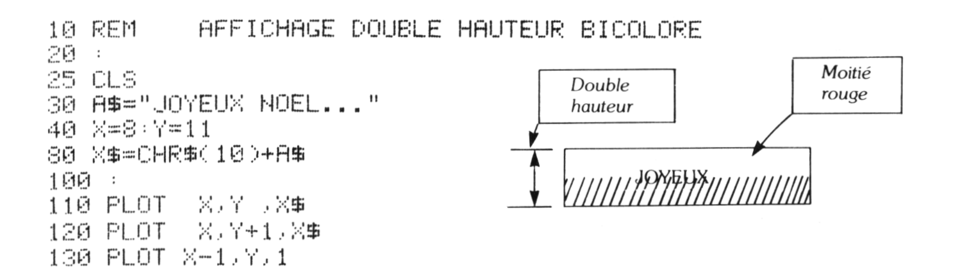
Affichage multicolore double hauteur

```
10 REM    AFFICHAGE MULTICOLORE DOUBLE HAUTEUR
20 :
25 CLS
30 A$="JOYEUX NOEL..."
40 X=8:Y=11
50 X$=CHR$(10)
60 FOR I=1 TO LEN(A$)
70 :C=I-INT(I/7)*7      :REM couleur
80 :X$=X$+CHR$(C)+MID$(A$,I,1)
90 NEXT I
100 :
110 PLOT X,Y,X$
120 PLOT X,Y+1,X$
```

Ce qui apparait à l'écran → JOYEUX ... *Double hauteur*

Affichage bicolore double hauteur

```
10 REM    AFFICHAGE DOUBLE HAUTEUR BICOLORE
20 :
25 CLS
30 A$="JOYEUX NOEL..."
40 X=8:Y=11
80 X$=CHR$(10)+A$
100 :
110 PLOT  X,Y ,X$
120 PLOT  X,Y+1,X$
130 PLOT X-1,Y,1
```



Affichage multicolore avec défilement

```
10 REM   AFFICHAGE MULTICOLORE
20 :
25 CLS
30 A$="ORIC POUR TOUS.."
40 :
50 FOR I=1 TO LEN(A$)
60 :C=I-INT(I/7)*7:
70 :X$=X$+CHR$(C)+MID$(A$,I,1)
80 NEXT I
90 :
100 PLOT 1,10,X$
110 REM--------------Rotation--------------
120 X$=RIGHT$(X$,1)+LEFT$(X$,LEN(X$)-1)
125 WAIT 10
130 GOTO 100
```

JOYEUX NOEL...

Affichage multicolore double hauteur avec défilement

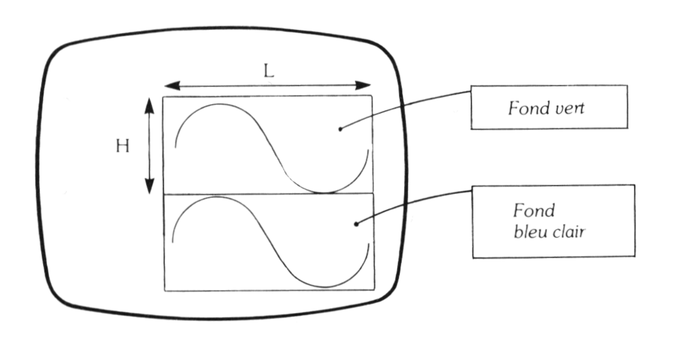
```
10 REM   AFFICHAGE MULTICOLORE DOUBLE HAUTEUR
15 REM   AVEC DEFILEMENT
20 :
25 CLS
30 A$="JOYEUX NOEL..."
40 X=8:Y=11
50 :
60 FOR I=1 TO LEN(A$)
70 :C=I-INT(I/7)*7      :REM couleur
80 :X$=X$+CHR$(C)+MID$(A$,I,1)
90 NEXT I
95 PLOT X-1,Y,10
96 PLOT X-1,Y+1,10
100 :
110 PLOT X,Y,X$
120 PLOT X,Y+1,X$
130 :
140 X$=RIGHT$(X$,2)+LEFT$(X$,LEN(X$)-2)
145 WAIT 10
150 GOTO 110
```

Double hauteur

JOYEUX ...

COULEURS BACKGROUND

Ci-dessous, nous changeons la couleur de fond (BACKGROUND) de 2 parties de l'écran. De la même façon, nous pourrions changer la couleur d'écriture.
Il suffirait d'écrire : FILL H, 1, C (au lieu de 16 + C).



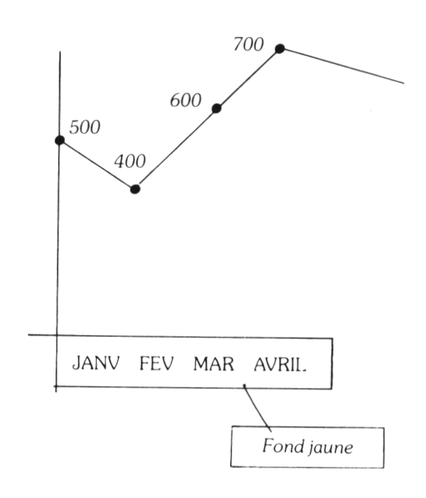
Fenêtre BACKGROUND

```
10 REM      FENETRE BACKGROUND
20 :
30 HIRES
35 INK1
39 REM------ Fenetre 1
40 X1=120:Y1=10 :H=50:L=90:C=2:GOSUB 200
45 REM-------- Fenetre 2
50 :
60 X1=120:Y1=100:H=50:L=90:C=6:GOSUB 200
65 REM------- Courbes
70 X0=170:Y0=35:GOSUB 80
75 X0=170:Y0=125:GOSUB 80
76 STOP
77 :
79 REM-----
80 FOR X=-PI TO PI STEP PI/15
90 :Y=SIN(X)
95 :CURSET X0+X*10,Y0+Y*20,1
100 NEXT X
110 RETURN
120 :
160 STOP
190 REM-------------- SPGM FENETRE
200 CURSET X1,Y1,0
210 FILL H,1,16+C
220 CURSET (X1+L),Y1,0
230 FILL H,1,16
240 RETURN
```

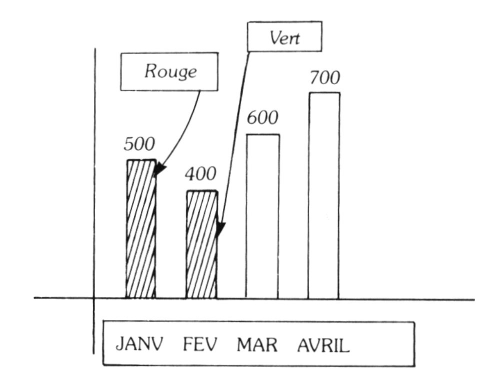
HISTOGRAMMES

Grâce aux couleurs, l'ORIC-1 permet une présentation attrayante des résultats. Nous présentons ci-dessous 2 types d'histogramme. Les valeurs sont contenues dans une table VNTE(). L'échelle est calculée automatiquement.

Avec DRAW :



Avec FILL :



Histogramme avec DRAW

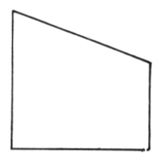
```
10 REM      HISTOGRAMME
20 :
25 NM=7
30 :
40 DATA JANV,500,    FEV,400
50 DATA MARS,600,    AVR,700
70 DATA MAI,600,     JUIN,500
90 DATA JUILL,700
100 FOR M=1 TO NM
110 :READ MOI$(M),VNTE(M)
120 NEXT M
130 :
140 REM------------------
150 HIRES:INK1:PAPER6
160 X0=20:Y0=150      :REM Depart axes
165 IX=30             :REM Intervalle
167 HECR=120          :REM Hauteur Y maxi
168 REM------------ Recherche maxi
170 MX=VNTE(1)
171 FOR M=2 TO NM
172 :IF VNTE(M)>MX THEN MX=VNTE(M)
173 NEXT M
175 ECH=HECR/MX
178 REM------------------ Axes
180 CURSET X0,Y0,1:DRAW NM*IX,0,1
200 CURSET X0,Y0,1:DRAW 0,-HECR,1
220 REM--------------- Affichage mois
225 FOR AM=40960+(Y0+3)*40 TO 40960+(Y0+20)*40 STEP 40
226 :POKE AM+2,16+3
227 NEXT AM
228 :
230 FOR M=1 TO NM
235 : CURSET X0+2 +IX*(M-1),Y0+10,0
236 : FOR I=1 TO LEN(MOIS$(M))
237 :   X=ASC(MID$(MOIS$(M),I,1))
238 :  CHAR X,0,1
239 :  CURMOV 6,0,3
240 :  NEXT I
245 NEXT M
280 REM----------- Courbe
290 CURSET X0,Y0-VNTE(1)*ECH,1
295 :
300 FOR M=2 TO NM
310 : DY=(VNTE(M)-VNTE(M-1))*ECH
320 : DRAW IX,-DY,1
330 NEXT M
340 REM------------ Affichage valeurs
350 FOR M=1 TO NM
360 :Y=Y0-VNTE(M)*ECH-15
365 :CURSET X0+2 +IX*(M-1),Y,0
370 :X$=STR$(VNTE(M))
380 :FOR I=2 TO LEN(X$)
390 : X=ASC(MID$(X$,I,1))
400 : CHAR X,0,1:CURMOV 6,0,3
410 :NEXT I
420 NEXT M
```

Histogramme avec FILL

```
10 REM HFILL HISTOGRAMME AVEC FILL
20 :
25 NM=7
30 :
40 DATA JANV,500,    FEV,400
50 DATA MARS,600,    AVR,700
70 DATA MAI,600,     JUIN,500
90 DATA JUILL,700
100 FOR M=1 TO NM
110 :READ MOI$(M),VNTE(M)
120 NEXT M
130 :
140 REM--------------------
150 HIRES:INK1
160 X0=20:Y0=150       :REM Depart axes
165 IX=30              :REM Intervalle
167 HECR=120           :REM Hauteur Y maxi
168 REM------------ Recherche maxi
170 MX=VNTE(1)
171 FOR M=2 TO NM
172 :IF VNTE(M)>MX THEN MX=VNTE(M)
173 NEXT M
175 ECH=HECR/MX
178 REM------------------ Axes
180 CURSET X0,Y0,1:DRAW NM*IX,0,1
200 CURSET X0,Y0,1:DRAW 0,-HECR,1
220 REM--------------- Affichage mois
225 FOR AM=40960+(Y0+3)*40 TO 40960+(Y0+20)*40 STEP 40
226 :POKE AM+2,16+3
227 NEXT AM
228 :
230 FOR M=1 TO NM
235 : CURSET X0+2 +IX*(M-1),Y0+10,0
236 : FOR I=1 TO LEN(MOIS$(M))
237 :    X=ASC(MID$(MOIS$(M),I,1))
238 :   CHAR X,0,1
239 :   CURMOV 6,0,3
240 :   NEXT I
245 NEXT M
280 REM----------- Courbe
300 FOR M=1 TO NM
310 : DY=         VNTE(M)*ECH
312 : CURSET X0+IX*(M-1)+4,Y0-DY,0
314 : FILL DY,1,16+M
315 : CURSET X0+IX*(M-1)+12,Y0-DY,0
320 : FILL DY,1,16
330 NEXT M
340 REM------------ Affichage valeurs
350 FOR M=1 TO NM
360 :Y=Y0-VNTE(M)*ECH-15
365 :CURSET X0+2 +IX*(M-1),Y,0
370 :X$=STR$(VNTE(M))
380 :FOR I=2 TO LEN(X$)
390 : X=ASC(MID$(X$,I,1))
400 : CHAR X,0,1:CURMOV 6,0,3
410 :NEXT I
420 NEXT M
```

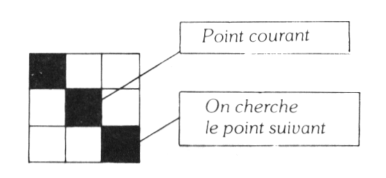
REMPLISSAGE D'UNE FIGURE

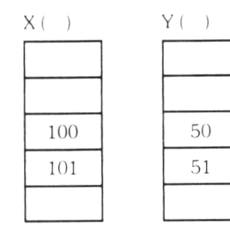
Le problème consiste à remplir une figure fermée déjà dessinée sur l'écran.



Parcours du périmètre :

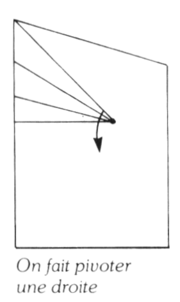
A l'aide de l'instruction **POINT (X,Y),** nous parcourons le périmètre de la figure et nous stockons tous les points dans 2 tables X() et Y().





Remplissage :

Il suffit de choisir un point à l'intérieur de la figure et de la faire pivoter autour de celui-ci.



On fait pivoter une droite

Nous pourrions calculer le périmètre de la figure en faisant :
Périmètre $= \sum$ **SQR(DX*DX+DY*DY)**

Remplissage d'une figure

```
10 REM REMPLISSAGE D'UNE FIGURE
20 :
22 DIM X(150),Y(150)
25 REM---------- Trace d'une figure pour essai
30 HIRES
40 CURSET 20,2,1
50 DRAW 20,10,1
55 DRAW 0,30,1
56 DRAW -20,0,1
57 DRAW 0,-40,1
70 REM------- Recherche premier point
80 FOR Y=0 TO 199
90 :FOR X=0 TO 239
100 : IF POINT(X,Y)=-1 THEN 150
110 :NEXT X
120 NEXT Y
130 PRINT "ECRAN VIERGE":STOP
140 :
150 PRINT "PREMIER POINT:";X,Y
160 REM================= Recherche de points
170 :
180 XC=X:YC=Y                  :REM Point courant
185 XD=X:YD=Y
190 :
200 FOR P=3 TO 1000
220 : FOR Y1=YC-1 TO YC+1    :REM Recherche point suivant
230 :  FOR X1=XC-1 TO XC+1
240 :   IF X1=XC THEN IF YC=Y1 THEN 270
250 :
255 :   IF POINT(X1,Y1)<>-1 THEN 270
256 :
260 :    FOR I=1 TO 3        :REM Retour arriere
262 :      IF X1=X(P-I) THEN IF  Y1=Y(P-I) THEN 270
265 :    NEXT I
267 :    XC=X1:YC=Y1:GOTO 310
270 :   NEXT X1
280 : NEXT Y1
290 :PRINT "FIN";:GOTO 700
300 :
310 :IF XC=X(3) THEN IF YC=Y(3) THEN 700
312 :X(P)=XC:Y(P)=YC
315 :CURSET XC,YC,0:CURSET XC,YC,1
320 NEXT P
330 :
690 :REM------ Remplissage
700 CX=30:CY=15          :REM Centre
705 FOR I=3 TO P-1
707 :DX=X(I)-CX:DY=Y(I)-CY
708 : IF DX=0 THEN IF DY=0 THEN 730
710 :CURSET CX,CY,1
720 :DRAW DX,DY,1
730 NEXT I
```

RECHERCHE D'UNE SOUS-CHAINE

Il est fréquent que nous recherchions la position d'une sous-chaîne dans une chaîne de caractères.



Dans le programme ci-dessous, l'opérateur pose la question **'DONNE-MOI LE PRODUIT DE 20 PAR 10'.** En recherchant les positions de **'PRODUIT DE'** et de **'PAR',** nous répondons à la question.

Recherche d'une sous-chaîne dans une chaîne

```
10 REM    RECHERCHE D'UNE SOUS-CHAINE
20 REM    DANS UNE CHAINE
30 :
40 CLS:INPUT "Votre question ";Q$
45 :
50 CH$=Q$:CCH$="PRODUIT DE":GOSUB 200
60 :
70 IF P=0 THEN 40
80 :
90 F1$=RIGHT$(Q$,LEN(Q$)-P-10):F1=VAL(F1$)
100 :
110 CH$=F1$:CCH$="PAR":GOSUB 200
120 :
130 IF P=0 THEN 30
140 :
150 F2$=RIGHT$(F1$,LEN(F1$)-P-3):F2=VAL(F2$)
160 :
170 PRINT:PRINT F1*F2
180 STOP
190 REM-------------- SPGM de recherche
200 L=LEN(CCH$)
210 :
220 FOR P=1 TO LEN(CH$)-L
230 :IF MID$(CH$,P,L)=CCH$ THEN 300
240 :NEXT P
250 P=0
260 :
300 RETURN
RUN

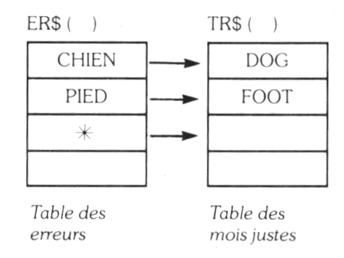
Votre question? DONNE  MOI LE PRODUIT DE 20 PAR 10

200
```

APPRENTISSAGE ANGLAIS AVEC PRIORITÉ

Nous choisissons un mot français au hasard dans la liste des DATAS. Nous l'affichons et demandons sa traduction anglaise.

En cas de mauvaise réponse, nous stockons l'erreur dans une table ER$() et lorsque 5 erreurs ont été commises, nous affichons les mots de la table des erreurs. Ainsi les mots mal connus sont présentés plus souvent.



Voisinage de la réponse

Nous pourrions vérifier si l'élève a seulement oublié une lettre (voir page 47).

Apprentissage anglais avec priorité

```
10 REM APANG     APPRENTISSAGE ANGLAIS AVEC PRIORITE
20 :
30 :
40 NMOT=17
45 CLS
50 :
60 DATA LIVRE,BOOK,          TETE,HEAD
70 DATA TABLE,TABLE,         LUNDI,MONDAY
80 DATA CHAT,CAT,            DENT,TOOTH.
90 DATA CHIEN,DOG,           NEZ,NOOSE
100 DATA MAISON,HOUSE
110 DATA VOITURE,CAR
120 DATA JAMBE,LEG
130 DATA PROBLEME,PROBLEM
140 DATA BOITE,BOX
150 DATA BRAS,ARM
160 DATA PIED,FOOT
170 DATA CHEF,BOSS
180 DATA CHAUSSURE,SHOE
290 :
300 IF NERR>5 THEN 500
310 :
320 RESTORE
325 X=INT(RND(1)*NMOT):IF XA=X THEN 325
327 XA=X
330 FOR I=1 TO X:READ FR$,ANG$:NEXT I
340 :
350 READ FR$,ANG$
360 :
370 PRINT:PRINT FR$;" ";
380 INPUT "Anglais ";R$
390 IF R$=ANG$ THEN PRINT:PRINT "OK":PRINT:GOTO 300
400 :
410 REM-----------Ajout erreur
420 FOR I=1 TO 100
430 :IF ER$(I)="" OR ER$(I)="*" THEN ER$(I)=FR$:TR$(I)=ANG$:GOTO 46
0
440 NEXT I
450 :
460 NERR=NERR+1
470 PRINT :PRINT "Erreur la bonne reponse est:";ANG$:PRINT
480 GOTO 300
```

```
490 REM------------- Tirage table des erreurs
500 IF NERR<3 THEN 300
504 :
505 X=INT(RND(1)*NERR)+1:IF X=XA THEN 505
506 IF ER$(X)="*" THEN 505
507 XA=X
510 FR$=ER$(X):ANG$=TR$(X)
520 PRINT:PRINT FR$;" ";
530 INPUT "Anglais ";R$
540 IF R$=ANG$ THEN PRINT:PRINT "OK":PRINT:ER$(X)="*":NERR=NERR-1:G
OTO 500
550 :
560 PRINT :PRINT "Erreur la bonne reponse est:";ANG$:PRINT
570 GOTO 500
```

Deuxième partie
ÉDUCATION

TRACÉ D'UN DESSIN DÉFINI EN DATA

Le programme ci-dessous trace une figure en reliant entre eux des points définis en DATA.

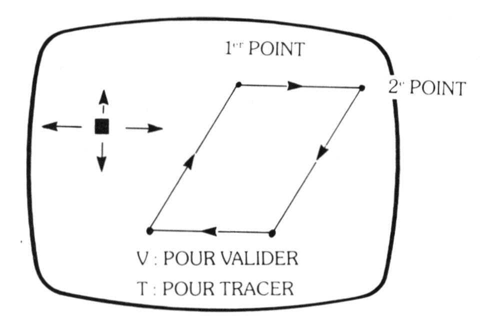
Tracé de dessin défini en data

```
10 REM  TD1   TRACE DE DESSIN DEFINI EN DATA
20 :
24 :
25 REM CARTE DE FRANCE
27 :
30 DATA 100,91
40 DATA 103,82,   106,76,   115,85
50 DATA 130,94,   148,100,  145,115
60 DATA 136,130,  142,130,  142,139
70 DATA 145,154,  148,160,  148,166
80 DATA 142,169,  136,172,  124,172
90 DATA 118,184,  109,184,  97,181
100 DATA 94,184,  82,175,   85,157
110 DATA 85,148,  85,139,   79,127
120 DATA 70,118,  64,118,   64,115
130 DATA 61,112,  61,109,   67,109
140 DATA 70,106,  73,106,   82,109
150 DATA 85,106,  82,97,    85,94
160 DATA 88,97,   94,97,   100,91
200 DATA 999,999
210 REM--------
215 HIRES:INK6
220 READ XA,YA
225 CURSET XA,YA,1
228 :
230 READ X,Y:IF X=999 THEN 290
240 :
250 DRAW X-XA,Y-YA,1
260 XA=X:YA=Y
270 GOTO 230
280 REM---------- CoPie imPrimante
290 LPRINT
300 FOR L=80 TO 199 STEP 6
305 :
310 :FOR C=20 TO 190 STEP 3
315 : F=0
320 : FOR I=0 TO 2
330 :   IF POINT(C+I,L  )=-1 THEN F=1
340 : NEXT I
350 : IF F=1 THEN LPRINT "*";:GOTO 360
355 : LPRINT " ";
360 :NEXT C
370 :LPRINT
375 :
380 NEXT L
```

TRACÉ D'UN DESSIN PAR SEGMENTS DE DROITES ET DIGITALISATION D'UN DESSIN

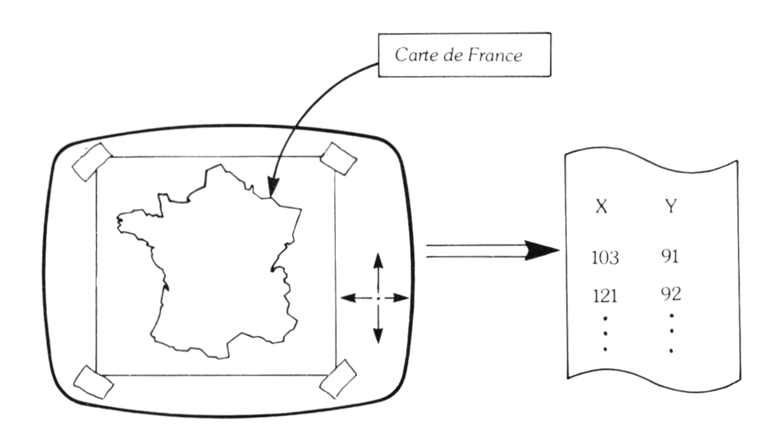
Le programme ci-dessous permet de déplacer un curseur à l'aide des 4 flèches →← ↑ ↓, de **'valider' des points** et de **tracer des droites** entre ces points.

- **Pour 'valider' un point,** nous appuyons sur 'V'.
- **Pour tracer une droite** entre le point courant et le point valide précédent, nous appuyons sur 'T'.



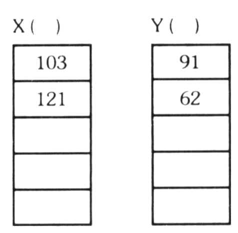
Le programme permet également de **'digitaliser'** un dessin :

- Nous collons sur l'écran une feuille transparente sur laquelle a été décalquée la carte de France par exemple.
- Nous déplaçons le curseur sur le périmètre de la figure et périodiquement, nous 'validons' les points. Les coordonnées X et Y des points s'affichent alors sur l'imprimante.



EXTENSIONS POSSIBLES :

- Nous pourrions stocker les points 'validés', dans 2 tables X() et Y().



- La suppression et l'insertion de points permettrait de corriger le dessin.
- Les tables pourraient être sauvegardées dans un fichier.

Le programme 'SIGN', ci-dessous, reproduit une signature 'digitalisée' avec le programme 'TDES'.

Digitalisation de dessin+tracé par segments

```
10 REM TDES  DIGITALISATION DE DESSIN+TRACE PAR SEGMENTS
12 :
20 :
25 HIRES:INK6
30 X=100:Y=100          :REM Coordonnees depart
40 :
70 :
80 C$=KEY$:IF C$<>"" THEN 130  :REM Attente frappe clavier
90 CURSET X,Y,1
100 CURSET X,Y,0
110 GOTO 80
120 :
130 IF T=1 THEN CURSET X,Y,1
140 :
150 C=ASC(C$)
160 IF C=8 THEN X=X-1
170 IF C=9 THEN  X=X+1
180 IF C=10 THEN  Y=Y+1
190 IF C=11 THEN  Y=Y-1
200 :
210 IF C$="E" THEN CURSET X,Y,0
220 IF C$="V" THEN CURSET X,Y,1:XA=X:YA=Y:GOSUB 500
225 IF C$="T" THEN CURSET XA,YA,1:DRAW X-XA,Y-YA,1:XA=X:YA=Y:GOSUB
500
230 T=0:IF POINT(X,Y)=-1 THEN T=1
240 GOTO 80
490 :
500 LPRINT X,Y:RETURN
```

Signature

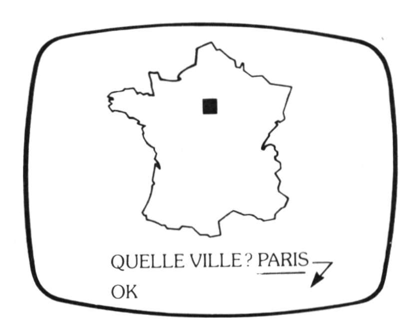
```
10 REM SIGN       SIGNATURE
20 :
24 :
30 DATA 127,101,     118,102,   106,103
40 DATA 94,104,      82,104,    73,104
50 DATA 65,103,    62,102,    61,101
60 DATA 62,99,     64,97,     68,95
70 DATA 74,93,    82,91,      94,90
80 DATA 102,91,   107,92,     110,93
90 DATA 112,94,   113,96,     112,98
100 DATA 110,99,  113,100,    116,101
110 DATA 118,104, 118,106,    117,109
120 DATA 114,111,  108,113,    102,115
130 DATA 94,117,   85,118,     73,118
140 DATA 66,117,   62,116,    61,114
150 DATA 62,112,   64,111,    68,109
200 DATA 999,999
210 REM--------
215 HIRES:INK6
220 READ XA,YA
225 CURSET XA,YA,1
228 :
230 READ X,Y:IF X=999 THEN END
240 :
250 DRAW X-XA,Y-YA,1
260 XA=X:YA=Y
270 GOTO 230
```



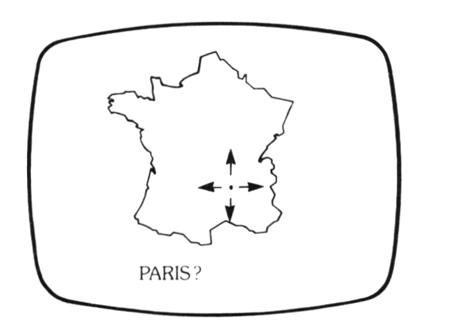
CARTE DE FRANCE

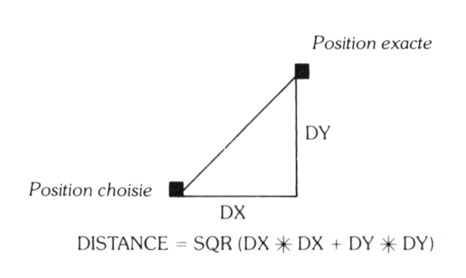
Le programme ci-dessous permet de tester les connaissances en géographie. Il existe 2 modes d'utilisation :

• **Nous présentons la position d'une ville et nous demandons le nom de la ville** (Mode 'IV1').



• **Nous demandons de situer la position d'une ville.** L'élève déplace un curseur et valide la position choisie par 'V'. Nous donnons à l'élève l'écart entre la position choisie et la position exacte. (Mode 'IV2')





• **Pour accélérer la vitesse de déplacement du curseur, tester la mémoire # 208.** (Voir page 13)

Carte de France

```
10 REM   CARTE DE FRANCE
15 :
18 DIM XV(20),YV(20),V$(20)
20 :
30 DATA 100,91
40 DATA 103,82,    106,76,    115,85
50 DATA 130,94,    148,100,   145,115
60 DATA 136,130,   142,130,   142,139
70 DATA 145,154,   148,160,   148,166
80 DATA 142,169,   136,172,   124,172
90 DATA 118,184,   109,184,   97,181
100 DATA 94,184,   82,175,    85,157
110 DATA 85,148,   85,139,    79,127
120 DATA 70,118,   64,118,    64,115
130 DATA 61,112,   61,109,    67,109
140 DATA 70,106,   73,106,    82,109
150 DATA 85,106,   82,97,     85,94
160 DATA 88,97,    94,97,    100,91
165 DATA 999,999
170 REM----------------------- VILLES
175 DATA 94,103,CAEN
176 DATA 109,106,PARIS
177 DATA 106,115,ORLEANS
178 DATA 97,115,LE MANS
180 DATA 73,118,LORIENT
182 DATA 67,112,BREST
184 DATA 97,133,POITIERS
186 DATA 103,145,LIMOGES
188 DATA 91,157,BORDEAUX
190 DATA 130,139,LYON
192 DATA 136,148,GRENOBLE
194 DATA 136,169,MARSEILLE
196 DATA 106,172,TOULOUSE
200 DATA 999,999,ZZZZ
210 REM--------
215 HIRES:INK3
220 READ XA,YA
225 CURSET XA,YA,1
228 :
230 READ X,Y:IF X=999 THEN 300
240 :
250 DRAW X-XA,Y-YA,1
260 XA=X:YA=Y
270 GOTO 230
290 REM---------------------------
295 V=0
300 READ X,Y,V$:IF X=999 THEN NV=V:GOTO 400
305 V=V+1
310 XV(V)=X:YV(V)=Y:V$(V)=V$
320 GOTO 300
```

```
390 REM======================== MENU =======
400 INPUT "MODE (IV1,IV2) ";M$
410 IF M$="IV1" THEN GOSUB 1000
415 IF M$="IV2" THEN GOSUB 1200
450 GOTO 400
460 :
465 :
470 :
475 :
480 :
485 :
490 :
495 :
990 REM==================== INTERROGATION VILLE ===
1000 V=INT(RND(1)*NV)+1
1010 CURSET XV(V),YV(V),1
1020 INPUT "QUELLE VILLE ";V$:IF V$="FIN" THEN RETURN
1030 IF V$=V$(V) THEN PRINT "OK":WAIT 200:GOTO 1100
1040 :
1050 PRINT "NON! C'EST ";V$(V):WAIT 200:GOTO 1100
1090 :
1100 CURSET XV(V),YV(V),0
1110 GOTO 1000
1120 REM================ INTERROGATION PAR VILLE ====
1200 X=100:Y=100
1203 :
1204 V=INT(RND(1)*NV)+1
1205 PRINT "OU EST SITUE ";V$(V)
1206 PRINT "UTILISER 4 FLECHES PUIS 'V'"
1210 :
1230 :
1280 C$=KEY$:IF C$<>"" THEN 1330
1290 CURSET X,Y,1
1300 CURSET X,Y,0
1310 GOTO 1280
1320 :
1330 IF T=1 THEN CURSET X,Y,1
1340 :
1350 C=ASC(C$)
1360 IF C=8 THEN X=X-3
1370 IF C=9 THEN X=X+3
1380 IF C=10 THEN Y=Y+3
1390 IF C=11 THEN Y=Y-3
1400 :
1410 IF C$="V" THEN 1600
1420 T=0:IF POINT(X,Y)=-1 THEN T=1
1430 GOTO 1280
1450 :
1600 DX=XV(V)-X:DY=YV(V)-Y
1610 D=SQR(DX*DX+DY*DY)
1620 PRINT "VOUS ETES A ";D*10;"KM"
1630 CURSET XV(V),YV(V),1
1635 WAIT 200
1640 GOTO 1204
```

TRACÉ DE DESSIN EN RELATIF AVEC ÉCHELLE

La carte de France du programme précédent est tracée à l'aide de coordonnées absolues.

Sur le programme ci-dessous, nous spécifions en DATAS les coordonnées relatives au point précédent. Ainsi nous pouvons agrandir ou réduire la taille du dessin.

Pour obtenir les coordonnées relatives à l'aide des coordonnées absolues, nous faisons :

```
30 DATA 100,91         :REM 1 er Point
40 DATA 103,82
50 DATA ......
200 DATA 999,999
210:
300 READ XA,YA         : 1er Point
310 :
320 READ X,Y:IF X=999 THEN STOP
330 :
340 DX=X-XA:DY=Y-YA
350 PRINT DX,DY
360 XA=X:YA=Y
370 GOTO 320

RUN

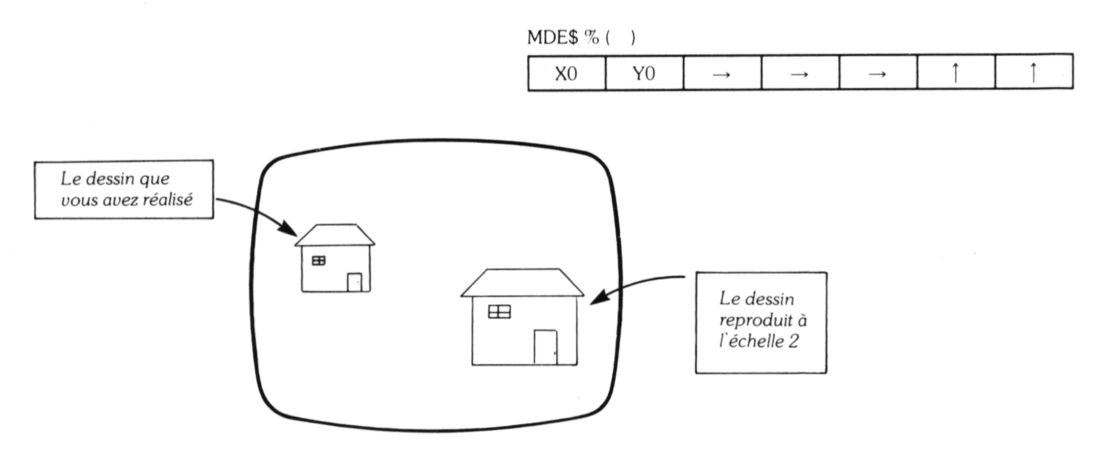
3  -9
3  -6
```

Tracé de dessin en relatif avec échelle

```
10 REM TD2     TRACE DE DESSIN EN RELATIF AVEC ECHELLE
20 :
30 DATA 100,30                       :REM 1 er Point
40 DATA 3,-9,    3,-6,    9,9
50 DATA          15,9,    18,6
60 DATA -3,15,   -9,15,   6,0
70 DATA 0,9,     3,15,    3,6
80 DATA 0,6,    -6,3,     -6,3
90 DATA -12,0,  -6,12,    -9,0
100 DATA -12,-3,  -3,3,     -12,-9
110 DATA 3,-18,   0,-9,   0,-9
120 DATA -6,-12,   -9,-9,   -6,0
130 DATA 0,-3,   -3,-3,  0,-3
140 DATA 6,0,    3,-3,     3,0
150 DATA 9,3,    3,-3,    -3,-9
160 DATA 3,-3,   3,3,   6,0
170 DATA 6,-6
200 DATA 999,999
210 :
300 HIRES:INK3
305 ECH=1.6                  :REM ECHELLE
310 READ XA,YA
320 CURSET XA,YA,1
330 :
400 READ DX,DY:IF DX=999 THEN STOP
410 DRAW DX*ECH,DY*ECH,1
415 :
417 XA=XA+DX*ECH:YA=YA+DY*ECH
418 CURSET XA,YA,1
420 GOTO 400
```

TÉLÉCRAN AVEC ÉCHELLE ET SAUVEGARDE

Nous utilisons ici le principe bien connu du **'TÉLÉCRAN'** :
En déplaçant un curseur à l'aide des 4 flèches ↑ ↓ ← →, nous représentons un dessin en laissant une **'tracé'** derrière le curseur.
Mais en plus, nous stockons dans une table MDES%() tous les déplacements. Ainsi, nous sommes capables de dupliquer le dessin réalisé dans une autre partie de l'écran avec une autre échelle (1, 2 ou 3).

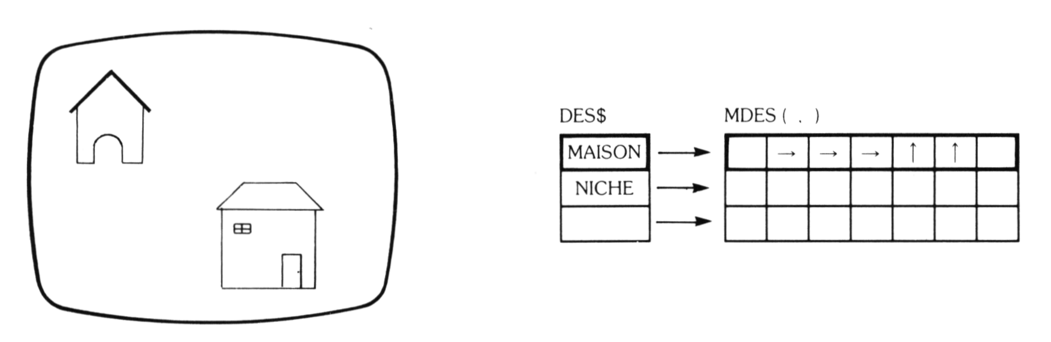


Il est prévu un mode pour sauvegarder la table MDES%() sur cassette. Ainsi, le dessin peut être réaffiché et amélioré. Si vous ne désirez pas sauvegarder les dessins, les instructions après 760 ne sont pas nécessaires.

AMÉLIORATIONS POSSIBLES :

Nous pourrions gérer plusieurs dessins :

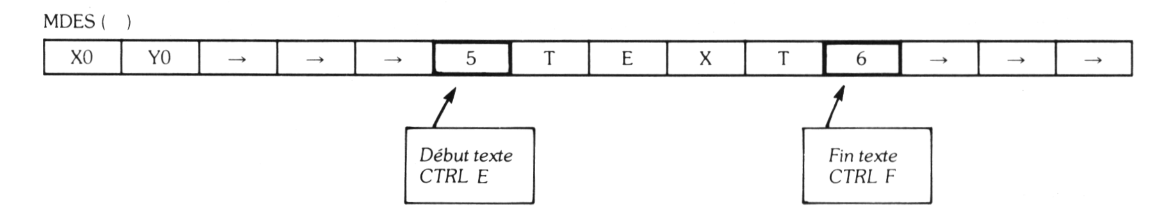
- Une table à 2 dimensions MDES(,) nous permettrait de gérer plusieurs dessins indépendants simultanément.



• Le résultat d'une composition de dessins pourrait être sauvegardée par une sauvegarde directe de la mémoire écran par : SCAVE «DES», A40960, E49000.

• Un dessin pourrait être effacé en utilisant MDES() (comme pour une duplication), mais avec CURSET X,Y,0 au lieu de CURSET X,Y,1

• L'affichage de texte se ferait en spécifiant à l'aide d'un code (CTRL E par exemple) que les caractères à suivre sont du texte.



• Pour accélérer la vitesse de déplacement du curseur, tester la mémoire # 208.
(Voir page 13)

• **On pourra ajouter :** `15 HIMÈM #97FF`

Télécran avec échelle et sauvegarde

```
10 REM SDES    TELECRAN AVEC ECHELLE ET SAUVEGARDE
15 :
16 CLS
17 DIM MDES%(500)
20 GOSUB 20000          ' POUR SAUVEGARDE SEULEMENT
50 :
60 CLS
70 PRINT "CDES : COMPOSE UN DESSIN"
75 PRINT "       UTILISER LE 4 FLECHES"
80 PRINT "        F POUR FIN DESSIN
90 PRINT "ADES : AFFICHE UN DESSIN COMPOSE"
92 PRINT "S    : SAUVEGARDE DESSIN
95 PRINT "L    : LECTURE DESSIN"
98 REM------------------------
100 PRINT: INPUT "MODE (CDES,ADES,S,L) ";M$
110 IF M$="L" THEN GOSUB 800
120 IF M$="S" THEN GOSUB 900
130 IF M$="CDES" THEN GOSUB 170
135 IF M$="SD" THEN GOSUB 3000
136 IF M$="LD" THEN GOSUB 4000
140 IF M$="ADES" THEN GOSUB 495
150 GOTO 100
160 REM================= COMPOSITION DESSIN =====
170 L=0
180 HIRES
185 PRINT "B:BAISSER L:LEVER F:FIN";
190 X=100:Y=100
200 MDES%(1)=X:MDES%(2)=Y
210 P=3
230 :
240 C$=KEY$:IF C$<>"" THEN 300
250 CURSET X,Y,1
260 CURSET X,Y,0
270 GOTO 240
300 IF T=1 THEN CURSET X,Y,1
310 :
314 IF C$="L" THEN L=2
315 IF C$="F" THEN RETURN
316 IF C$="E" THEN L=1
317 IF C$="B" THEN L=0
318 :
319 IF L<2 THEN CURSET X,Y,1-L
320 C=ASC(C$)
330 MDES%(P)=C:P=P+1
350 GOSUB 400
370 T=0:IF POINT(X,Y)=-1 THEN T=1
380 :
390 GOTO 240
```

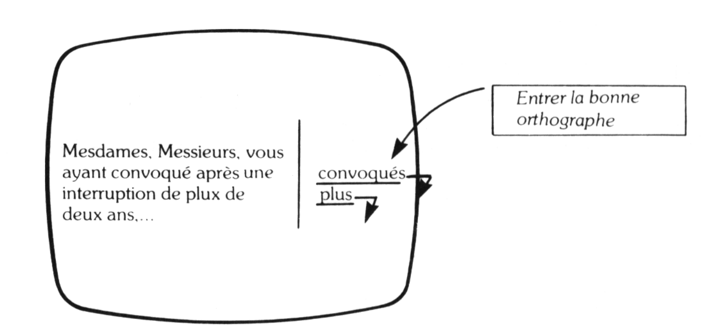
```
400 REM---------
410 IF C=8 THEN IF X>1 THEN X=X-1
420 IF C=9 THEN IF X<239 THEN X=X+1
430 IF C=10 THEN IF Y<199 THEN Y=Y+1
440 IF C=11 THEN IF Y>1 THEN Y=Y-1
450 RETURN
490 REM=============== AFFICHAGE DESSIN MDES%( ) =====
495 HIRES
500 PRINT "ECHELLE (1,2,3) ";:GET E$:PRINT E$
502 ECH=VAL(E$):IF ECH<1 OR ECH>3 THEN RETURN
505 :
507 PRINT "COULEUR ENCRE (0,1,2,..,7) ";:GET E$
508 PRINT E$:IF VAL(E$)<>0 THEN INK VAL(E$)
510 PRINT "EFFACEMENT (O/N) ";:GET R$
512 PRINT R$:IF R$="O" THEN HIRES
515 INPUT "X ";X:IF X=0 THEN X=MDES%(1)
517 INPUT "Y ";Y:IF Y=0 THEN Y=MDES%(2)
518 GOSUB 535
519 GOTO 500
520 :
535 L=0
540 FOR P=3 TO 300
550 : C=MDES%(P):IF C=0 THEN 640
560 :IF C=69 THEN L=1
570 :IF C=66 THEN L=0
575 :IF C=76 THEN L=2
580 :FOR J=1 TO ECH
590 : IF L=2 THEN 610
600 : CURSET X,Y,1-L
610 : GOSUB 400
620 :NEXT J
630 NEXT P
640 RETURN
790 REM------------- LECTURE CASSETTE
800 PRINT "APPUYER SUR <PLAY> PUIS <RETURN>"
810 GET X$
815 TEXT
820 PRINT "LECTURE CASSETTE"
830 POKE #67,0       ' 1:300 BAUDS
840 CALL 1027,MDES%
850 PRINT:PRINT "LECTURE TERMINEE":PRINT
855 HIRES: X=MDES%(1):Y=MDES%(2):GOSUB 535
860 GOTO 240
890 REM================== SAUVEGARDE CASSETTE ======
900 TEXT
910 PRINT "APPUYER SUR <RECORD> PUIS <RETURN> "
920 GET X$
930 PRINT "SAUVEGARDE":PRINT
940 POKE #67,0       ' 1:300 BAUDS
950 CALL 1024,MDES%
960 PRINT:PRINT "SAUVEGARDE TERMINEE":PRINT
970 RETURN
```

Routine sauvegarde/lecture table

```
19990 REM------------------- ROUTINE SAUVEGARDE/LECTURE TABLE --
19995 REM   Cf revue ORIC septembre 1983 ou Programme ORIC BASE
19996 :
20000 CLS:PRINT "ATTENDEZ":PRINT
20005 A=#B800:READ D$
20010 FOR I=1 TO LEN(D$) STEP 2
20020 : V=VAL("#"+MID$(D$,I,2)):POKE A,V:A=A+1:NEXT I
20030 READ D$:IF D$<>"Z" THEN 20010
20040 DOKE #400,#0A4C:DOKE #402,#4CB8:DOKE #404,#B858
20045 RETURN
20050 DATA 55555555233944363855200BB90820D6B820BAE6A92520C6E5A53320
C6E5A53420
20060 DATA C6E520EEB820A7E524281003203 5B82004E82860A000B101F017AAA0
02B10199D0
20070 DATA 0088D0F8E8CAF008B1D120C6E5C8D0F520C3B890DE602095D5200BB9
0820D6B820
20080 DATA 96E62030E6C925D0F92030E685332030E68534A002B1CEC533C8B1CE
E534B00620
20090 DATA 04E84C83C420EEB820EBE424281003209BB82004E82860A000B101F0
1C20F0D4AA
20100 DATA E8A000CAF0082030E691D1C8D0F5A002B9D000910188D0F820C3B890
D96018A903
20110 DATA650185 01A89002E602A502C461E5626020CAE62018B9A003B1CEAA88B
1CEE901B0
20120 DATA01CA85338634601 8A5CE65338561A5CF65348562A004B1CE20F6D1855
F84608501
20130 DATA8402602 0E800C92CF0034CE4CF4CE200A20020E800862785B420E8002
086D1B006
20140 DATA2004E84CE4CFA2008628862920E20090052086D1900BAA20E20090FB2
086D1B0F6
20150 DATAC924D006A9FF8528D00CC925D00FA980852905B485B48A0980AA20E20
086B5A69E
20160 DATAA59F86CE85CFC5A1D004E4A0F01FA000B1CEC8C5B4D006A5B5D1CEF00
EC8B1CE18
20170 DATA65CEAAC8B1CE65CF90D738602004E8A22A4C85C455
20180 DATA Z
```

ORTHOGRAPHE

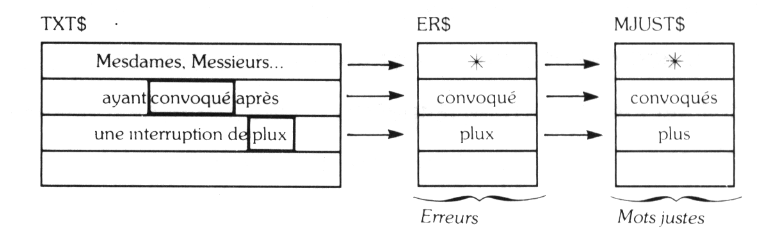
Nous présentons à l'élève un texte dans lequel figurent des fautes d'orthographe. En face des lignes contenant des fautes,8 l'élève doit frapper le mot erroné en l'orthographiant correctement.



Nous avons prévu **2 niveaux de difficulté :**

- Les mots mal orthographiés sont distingués par une couleur différente de celle du texte.
- Les mots mal orthographiés ne sont pas repérés.

La 'dictée' définie en DATA est lue dans 3 tables.



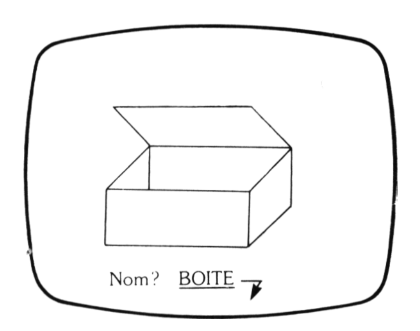
Orthographe

```
10 REM ORTHO      ORTHOGRAPHE
15 E$=CHR$(27)
17 INK4
20 :
30 DATA "Mesdames,Messieurs,vous",          *,       *
40 DATA "ayant convoque apres",             convoque,convoques
50 DATA "une interruption de plux",        plux,    plus
60 DATA "de quinse ans pour",              quinse, quinze
70 DATA "m'entretenir avec vous",          *,       *
80 DATA "de ce que le tamps ecoule",       tamps,   temps
90 DATA "ecoule a apporte de neuf",        *,       *
100 DATA *,*,*
250 FOR LG=1 TO 20
260 :READ TXT$(LG),ERR$(LG),MJUST$(LG)
270 :IF TXT$(LG)="*" THEN NL=LG:GOTO 290
280 NEXT LG
290 REM===================== MENU ==========
300 CLS
310 PRINT SPC(3);"AT  : Affichage texte"
320 PRINT SPC(3);"ATF : Affichage texte avec fautes"
400 PRINT:INPUT "Mode ";M$
410 IF M$="ATF" THEN GOSUB 995
420 IF M$="AT" THEN GOSUB 500
490 GOTO 300
500 REM================== AFFICHAGE TEXTE SANS ERREURS ==
510 CLS:PRINT
520 FOR LG=1 TO NL
530 :PRINT TXT$(LG)
540 NEXT LG
550 :
560 GOSUB 1205
570 PRINT:PRINT "Appuyez sur <RETURN> ":GET X$
580 RETURN
990 REM=================== AFFICHAGE TEXTE ET ERREURS  ===
995 CLS:PRINT
1000 FOR LG=1 TO NL
1010 :LIG$=TXT$(LG):ERR$=ERR$(LG):MJUST$=MJUST$(LG):GOSUB 3000
1014 :IF P=0 THEN PRINT LIG$:GOTO 1050
1015 :
1020 :G$=LEFT$(LIG$,P-1):D$=RIGHT$(LIG$,LEN(LIG$)-P-L+1)
1030 :
1040 :PRINT G$;" ";E$;"B";ERR$;E$;"D";D$
1045 :
1050 NEXT LG
1100 GOSUB 1205
1110 :
1120 PRINT:PRINT "Appuyez sur <RETURN> ":GET X$
1130 RETURN
```

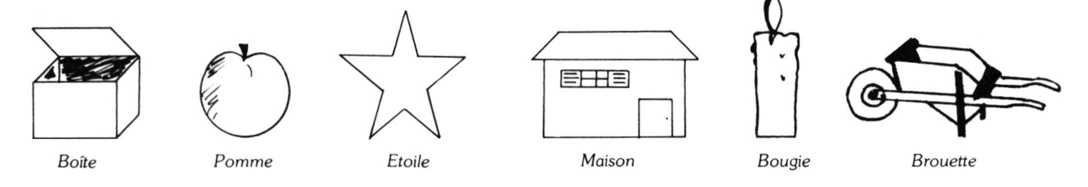
```
1200 REM=============== VERIF ERREURS
1205 PRINT CHR$(20);       :REM minuscules
1207 PLOT 5,23,"Entrez les corrections"
1208 NER=0
1210 FOR LG=1 TO NL
1220 : XL=29:YL=LG
1230 : GOSUB 4000
1240 : IF MJUST$(LG)="*"  AND LIG$="" THEN 1300
1245 : IF MJUST$(LG)<>"*" THEN 1250
1247 : PLOT 5,21,"Pas d'erreur!":WAIT 150:PLOT 5,21,"              "
:GOTO 1300
1249 :
1250 : IF MJUST$(LG)<>LIG$ THEN PING:NER=NER+1:GOTO 1220
1300 NEXT LG
1305 PRINT CHR$(20);         :REM Majuscules
1307 E$=STR$(NER)+" Erreur(s)":PLOT 5,21,E$
1310 RETURN
2990 REM----
2991 :
2992 :
2993 :
2994 :
3000 L=LEN(ERR$)
3010 FOR P=1 TO LEN(LIG$)-L+1
3020 :IF ERR$=MID$(LIG$,P,L) THEN RETURN
3030 NEXT P
3040 P=0:RETURN
3990 REM------------- INPUT AVEC KEY$
4000 LIG$=""
4010 :
4020 L=LEN(LIG$):PLOT XL+L,YL,130
4030 :
4040 C$=KEY$:IF C$="" THEN 4040
4045 C=ASC(C$)
4050 IF C=13 THEN PLOT XL+L,YL," ":RETURN
4060 IF C<>127 THEN 4100
4070 :
4080 IF L>0 THEN LIG$=LEFT$(LIG$,L-1):PLOT XL+L,YL," ":GOTO 4020 EL
SE 4040
4090 :
4100 IF C<32 THEN PING:GOTO 4040
4110 LIG$=LIG$+C$
4120 PLOT XL,YL,LIG$
4130 GOTO 4020
```

APPRENTISSAGE DE MOTS

Nous affichons une silhouette d'objet et nous demandons le nom de cet objet. Si une seule lettre de la réponse est oubliée (ou fausse), nous affichons la bonne orthographe du mot.



Nous avons représenté six objets :



Ces objets sont dessinés par segments de droites (voir digitalisation d'un dessin).

Apprentissage de mots

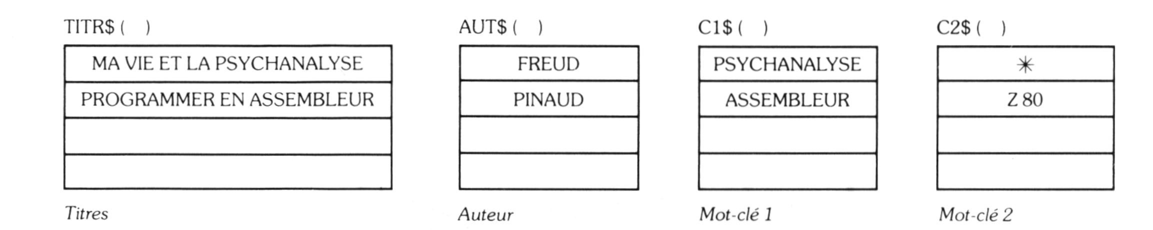
```
10 REM APMOT  APPRENTISSAGE DE MOTS
20 :
25 NMOT=6               ' Nombre de mots
30 :
90 REM---- BOITE
100 DATA           86,100,   86,87
110 DATA 107,87,  107,100,  86,100
120 DATA 86,87,   91,80,    111,80
130 DATA 107,87,  107,100,  112,94
140 DATA 112,80,  107,67,   87,67
142 DATA 92,80,   999,999
145 DATA BOITE
150 :
190 REM----- POMME
200 DATA 95,75,   99,74,     104,75
210 DATA 109,77,  112,81,    113,86
220 DATA 113,92,  111,98,    107,102
230 DATA 102,105,   96,106,   91,106
240 DATA 85,105,    80,103,   76,96
250 DATA 76,88,     78,81,     84,75
260 DATA 88,75,     93,77,     94,72
280 DATA 95,67,     999,999
285 DATA POMME
290 REM------- ETOILE
300 DATA 110,70,   117,86,    134,86
310 DATA 122,100,  125,121,   110,108
320 DATA 95,121,   98,102,    86,91
330 DATA 104,89,   109,70,    999,999
340 DATA ETOILE
350 REM------ BOUGIE
360 DATA 101,68,   101,128,   119,128
370 DATA 119,70,   101,68,    109,69
380 DATA 107,64,   107,59,    108,54
390 DATA 110,49,   112,47,    114,50
400 DATA 115,53,   115,58,    114,62
405 DATA 112,66,   110,69
410 DATA 999,999
420 DATA BOUGIE
430 REM------ MAISON
440 DATA 79,59,   110,59,  118,69
450 DATA 72,69,   78,59,   72,69
460 DATA 75,69,   75,105,  109,105
470 DATA 109,87,  101,87,  101,105
480 DATA 114,105, 114,69,  0,0
490 DATA 93,79,   83,79,   83,95
500 DATA 93,95,   93,79,   88,79
510 DATA 88,95,   93,95,   93,87
520 DATA 83,87,   999,999
540 DATA MAISON
```

```
550 REM----------- BROUETTE
560 DATA 75,107,  82,107,  86,113
570 DATA 83,121,  76,121,  72,115
580 DATA 75,107,  83,121,  86,114
585 DATA 73,114
590 DATA 75,107,  82,107,  76,120
595 DATA 0,0
600 DATA 86,113, 118,113, 122,120
610 DATA 119,113,  129,95,   140,95
620 DATA 72,95,     86,113
630 DATA 999,999
640 DATA BROUETTE
885 :
886 :
888 :
889 :
890 REM================== Lecture dessin au hasard ===
900 P=INT(RND(1)*NMOT):IF P=AP THEN 900
905 AP=P:RESTORE
907 IF P=0 THEN 1000
910 FOR I=1 TO P
920 : READ X,Y:IF X=999 THEN 930 ELSE 920
930 : READ MOT$
940 NEXT I
990 REM---------------- Affichage
1000 HIRES:INK6
1010 READ XA,YA
1020 CURSET XA,YA,1
1030 :
1050 READ X,Y:IF X=999 THEN 1100
1055 IF X=0 THEN READ X,Y:DRAW X-XA,Y-YA,3:GOTO 1070
1060 DRAW X-XA,Y-YA,1
1065 :
1070 XA=X:YA=Y
1080 GOTO 1050
1090 REM----------------- Reponse
1100 READ MOT$
1110 INPUT "Nom ";R$
1115 IF R$=MOT$ THEN PRINT "OK":GOTO 1200
1120 :
1125 GOSUB 1995
1130 :
1140 IF R= LEN(MOT$)-1 THEN PRINT "La bonne orthographe est:";MOT$:
GOTO 1200
1150 PRINT "La bonne reponse est:";MOT$
1190 :
1200 WAIT 200
1210 GOTO 900
1990 REM--------------------- Voisinage de la reponse ------
1995 R=0                        ' Nombre de bonnes lettres
2000 FOR I=1 TO LEN(R$)
2005 : C$=MID$(R$,I,1)
2010 : FOR J=1 TO LEN(MOT$)
2020 :    IF C$=MID$(MOT$,J,1) THEN R=R+1:GOTO 2050
2030 : NEXT J
2050 NEXT I
2060 :
2100 RETURN
```

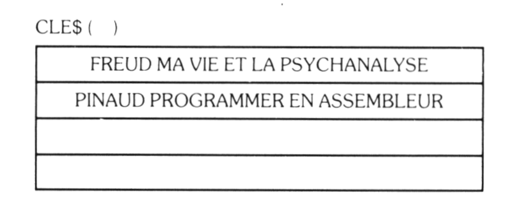
Troisième partie
GESTION

BIBLIOTHÈQUE EN DATA

Des titres d'ouvrages sont définis en 'DATA'. Afin de les éditer triés par titre, par auteur et par mot-clé, nous transférons les 'DATAS' dans des tables.



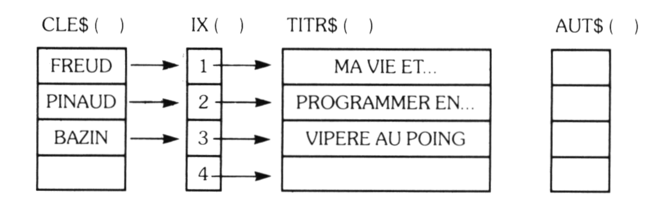
La méthode la plus simple pour obtenir la liste triée par auteur consiste à faire :



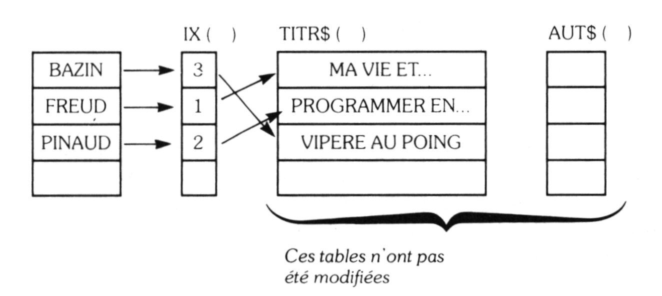

```
500 FOR LV=1 TO NLV
510  CLE$(LV)=AUT$(LV)+" "+TITR$(LV)
520 NEXT LV
530 :
540 REM  TRI DE CLE$( )
550 :
560 REM  EDITION DE CLE$( )
```

Cette façon de procéder présente 2 inconvénients :
- La table CLE$ occupe une place importante.
- Les résultats ne sont pas cadrés.

Aussi, procédons-nous autrement :
- Nous remplissons une **'table des Clés'** CLE$() avec les clés à trier (noms d'auteur sur l'exemple). Dans une **'table d'index'** IX(), nous rangeons les numéros de ligne (1, 2, 3...).



• Nous trions les tables CLE$() et IX()



• Après le tri, il suffit de lire la table IX() pour éditer les ouvrages dans l'ordre des auteurs.

```
FOR LV=1 TO NLV
 X=IX(LV)
 PRINT AUT$(X),TITR$(X)
NEXT X
```

Bibliothèque en DATA

```
10 REM BI       BIBLIOTHEQUE EN DATA
20 :
25 DIM TITR$(100),AUT$(100),C1$(100),C2$(100)
30 DIM CLE$(100),IX(100)
40 :
50 DATA MA VIE ET LA PSYCHANALYSE,FREUD,     PSYCHANALYSE, *
60 DATA VIPERE AU POING,           BAZIN,     ROMAN,        *
70 DATA PROGRAMMER EN ASSEMBLEUR, PINAUD,    ASSEMBLEUR,   Z80
80 DATA BASIC ET SES FICHIERS,    BOIBOI,    BASIC,        FICHIER
90 DATA LE BASIC PAR LA PRATIQUE, LAMOITIER, BASIC,        *
190 DATA *
200 REM============ LECTURE DATAS EN TABLE ===========
210 FOR LV=1 TO 200
220 :READ TIT$(LV):IF TIT$(LV)="*" THEN NLV=LV-1:GOTO 270
230 :READ AUT$(LV),C1$(LV),C2$(LV)
240 NEXT LV
270 :
300 REM====================== MENU =======
310 CLS:PRINT "Modes:":PRINT
320 PRINT SPC(3);"LTITRE:Liste par titre"
330 PRINT SPC(3);"LCLE  :Liste par mot-cle"
360 :
370 PRINT:INPUT "Mode ";M$
380 IF M$="LTITRE" THEN GOSUB 500
390 IF M$="LAUT" THEN GOSUB 1000
400 IF M$="LCLE" THEN GOSUB 1500
450 GOTO 310
490 REM===================== LISTE TRIEE PAR TITRE ===========
495 :
500 LPRINT:LPRINT "LISTE TRIEE PAR TITRE":LPRINT:LPRINT
505 :
510 FOR LV=1 TO NLV
520 :CLE$(LV)=TITR$(LV):IX(LV)=LV
530 NEXT LV
540 :
550 NCL=NLV:GOSUB 2000           :REM Appel tri de CLE$()
560 REM--------------Edition
570 FOR LV=1 TO NLV
580 : X=IX(LV)
590 :LPRINT TITR$(X);TAB(13+35);
600 :LPRINT AUT$(X)
650 NEXT LV
660 :
670 RETURN
990 REM================== LISTE PAR AUTEUR ====================
995 :
1000 LPRINT:LPRINT "LISTE TRIEE PAR AUTEUR":LPRINT:LPRINT
1010 FOR LV=1 TO NLV
1020 :CLE$(LV)=AUT$(LV):IX(LV)=LV
1030 NEXT LV
1040 :
1050 NCL=NLV:GOSUB 2000      :REM Appel tri cle$()
```

```
1060 REM---------Edition
1070 FOR LV=1 TO NLV
1080 : X=IX(LV)
1090 :LPRINT AUT$(X);TAB(13+15);
1100 :LPRINT TITR$(X)
1105 NCL=0
1110 NEXT LV
1120 :
1130 RETURN
1131 :
1132 :
1133 :
1134 :
1490 REM================== LISTE TRIEE PAR MOT-CLE =============
1495 :
1500 LPRINT:LPRINT "LISTE TRIEE PAR MOT CLE":LPRINT:LPRINT
1505 NCL=0                    ' Nombre de cles
1510 FOR LV=1 TO NLV
1512 :
1515 :IF LEFT$(C1$(LV),1) <>"*" THEN NCL=NCL+1:CLE$(NCL)=C1$(LV):IX
(NCL)=LV
1517 :IF LEFT$(C2$(LV),1) <>"*" THEN NCL=NCL+1:CLE$(NCL)=C2$(LV):IX
(NCL)=LV
1530 NEXT LV
1540 :
1550 GOSUB 2000               ' Appel tri cle$()
1560 :
1570 FOR LV=1 TO NCL
1580 : X=IX(LV)
1585 :IF CLE$(LV-1)<>CLE$(LV) THEN LPRINT:LPRINT CLE$(LV):LPRINT
1590 : LPRINT TAB(13+5);
1600 :LPRINT TITR$(X);TAB(13+50);
1605 :LPRINT AUT$(X)
1610 NEXT LV
1620 :
1630 RETURN
1990 REM================= TRI SHELL==========
2000 ECART=NCL
2010 :
2015 ECART=INT(ECART/2):IF ECART<1 THEN RETURN
2017 :
2020 IV=0
2030 FOR I=1 TO NCL-ECART
2040 :J=I+ECART
2050 :IF CLE$(J)=>CLE$(I) THEN 2080
2060 :X$=CLE$(I):CLE$(I)=CLE$(J):CLE$(J)=X$
2070 :X=IX(I):IX(I)=IX(J):IX(J)=X:IV=1
2080 NEXT I
2090 IF IV=1 THEN 2020 ELSE 2015
```

```
LISTE TRIEE PAR TITRE


BASIC ET SES FICHIERS                   BOIBOI
LE BASIC PAR LA PRATIQUE                LAMOITIER
MA VIE ET LA PSYCHANALYSE               FREUD
PROGRAMMER EN ASSEMBLEUR                PINAUD
VIPERE AU POING                         BAZIN


LISTE TRIEE PAR MOT CLE



ASSEMBLEUR

     PROGRAMMER EN ASSEMBLEUR                          PINAUD

BASIC

     LE BASIC PAR LA PRATIQUE                          LAMOITIER
     BASIC ET SES FICHIERS                             BOIBOI

FICHIER

     BASIC ET SES FICHIERS                             BOIBOI

PSYCHANALYSE

     MA VIE ET LA PSYCHANALYSE                         FREUD

ROMAN

     VIPERE AU POING                                   BAZIN

Z80

     PROGRAMMER EN ASSEMBLEUR                          PINAUD
```

```
10 REM    INDEX LIVRE
20 :
30 DIM CLE$(100),PG$(100)
40 :
50 DATA P22,INPUT,    P24,GOTO
70 DATA P27,IF,       P31,FOR
90 DATA P42,LEN,      P42,LEFT$
110 DATA P43,RIGHT$, P44,MID$
130 DATA P45,ASC,P46,CHR$,P47,STR$,P47,VAL
140 DATA P49,CONT,P52,GOSUB,P52,RETURN
290 DATA *,*
300 REM------------------- LECTURE DATAS DANS PG$( ) ET CLE$( )---
310 FOR I=1 TO 100
320 : READ PG$(I),CLE$(I)
330 : IF PG$(I)="*" THEN NM=I-1:GOTO 360
340 NEXT I
350 REM------------------------------------ TRI ---
360 IV=0
370 FOR I=1 TO NM-1
380 :IF CLE$(I+1)=>CLE$(I) THEN 400
390 :X$=CLE$(I):CLE$(I)=CLE$(I+1):CLE$(I+1)=X$:IV=1
395 :X$=PG$(I):PG$(I)=PG$(I+1):PG$(I+1)=X$
400 NEXT I
410 IF IV=1 THEN 360
420 REM------------------------------------ EDITION ----
430 :
500 FOR I=1 TO NM
510 :IF LEFT$(CLE$(I-1),1)= LEFT$(CLE$(I),1) THEN 520
515 :LPRINT:LPRINT LEFT$(CLE$(I),1):LPRINT
520 :LPRINT TAB(13+3);CLE$(I);TAB(13+12);PG$(I)
530 NEXT I
```

```
A

   ASC       P45

C

   CHR$      P46
   CONT      P49

F

   FOR       P31

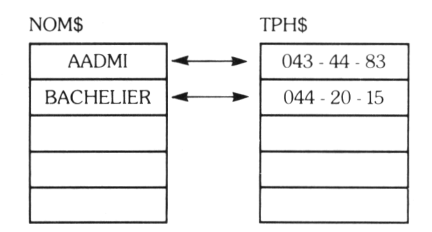
G

   GOSUB     P52
   GOTO      P24

I
```

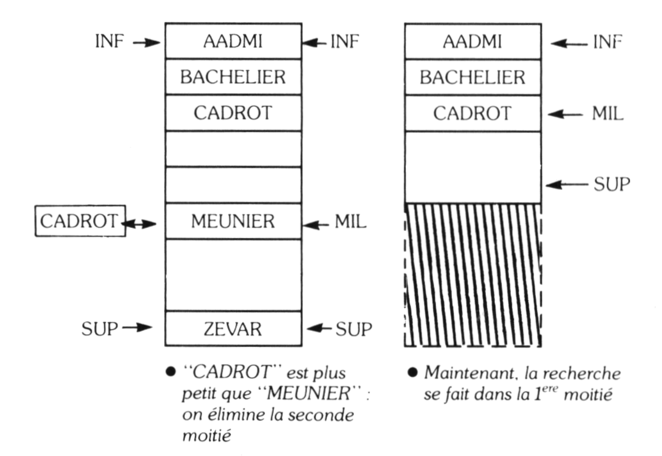
ANNUAIRE TÉLÉPHONIQUE (AVEC RECHERCHE DICHOTOMIQUE)

Une recherche **'séquentielle'** dans une table devient longue dès que le nombre d'éléments atteint 500. Si la table est **triée,** on peut retrouver très rapidement un élément par **'dichotomie'.**



Le principe est le suivant :
On compare le nom cherché ('CADROT' par exemple au nom du milieu de la table.
- S'il est plus petit, on en conclut qu'il ne peut être dans la seconde moitié de la table.
- S'il est plus grand, on en conclut qu'il ne peut être dans la première moitié.

Ainsi, nous divisons par 2 l'intervalle de recherche et en procédant de la même façon successivement, nous retrouvons très vite le nom cherché.



- *"CADROT" est plus petit que "MEUNIER" : on élimine la seconde moitié*
- *Maintenant, la recherche se fait dans la 1ère moitié*

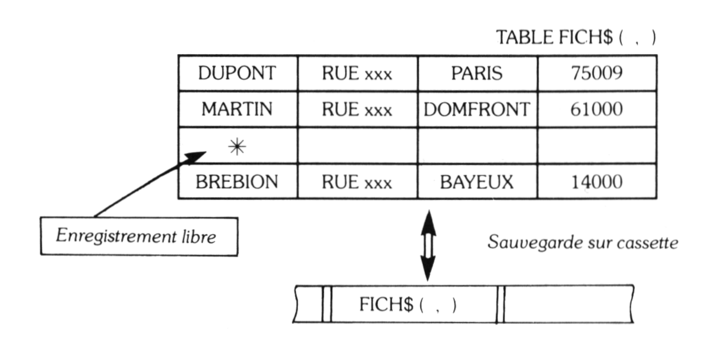
Annuaire

```
10 REM ANU       ANNUAIRE
20 :
30 :
40 DATA AADMI,   043.07.83,     ABIVEN, 043.07.78
50 DATA ABOAF,   043.60.97,     ABOU,    043.00.19
60 DATA ADAM,    043.62.66,     AGUERRE,  044.06.50
100 DATA BACHELIER,043.35.96, BAESLER,  043.17.34
130 DATA CADROT, 043.18.18,    CAHANIN,  043.01.31
150 DATA DA COSTA,043.30.92,    DA ROCHAS,044.23.50
160 DATA DABO,   044.05.58,    DABOUDET, 043.15.13
980 DATA *,*
990 REM---------------------- Lecture DATAS -----------------------
-
995 DIM NOM$(500),TPH$(500)
996 :
997 CLS
1000 FOR I=1 TO 1000
1010 :READ NOM$(I),TPH$(I)
1020 :IF NOM$(I)="*" THEN NN=I-1:GOTO 1100
1022 :PRINT NOM$(I)
1025 :IF I>1 THEN IF NOM$(I)<NOM$(I-1) THEN PRINT:PRINT "Non trie":
STOP
1030 NEXT I
1040 STOP
1100 REM------------- Recherche dichotomique ----
1110 PRINT:INPUT "Nom ";NOM$
1120 :
1130 INF=1:SUP=NN
1140 :
1150 MIL=INT((INF+SUP)/2)
1160 IF INF>SUP THEN PRINT:PRINT "N'existe pas":PRINT:GOTO 1110
1170 :
1180 IF NOM$=NOM$(MIL) THEN PRINT:PRINT NOM$(MIL);" ";TPH$(MIL):PRI
NT:GOTO 1110
1190 :
1200 IF NOM$<NOM$(MIL) THEN SUP=MIL-1 ELSE INF=MIL+1
1210 GOTO 1150
```

FICHIER D'ADRESSES

Sur l'exemple précédent, les informations (en 'DATA') ne peuvent être changées qu'en modifiant le programme.

Le programme ci-dessous permet d'introduire et de modifier des données indépendantes du programme. Elles sont temporairement stockées dans une table à 2 dimensions FICH$(,). Elles sont ensuite sauvegardées sur cassette.



Remarque : Ce programme peut être utilisé en définissant les adresses en DATA. Il faut dans ce cas transférer les DATAS dans la table FICH$(,)

Tri-sélection

Pour obtenir la liste triée des personnes d'une seule ville, il suffit de sélectionner les noms ainsi.

```
5000 INPUT "Quelle ville ";V$
5025 IF V$<>FICH$(F,3) THEN 5040
```

Tri multicritères

La liste alphabétique des clients dans l'ordre des villes s'obtient en faisant :

```
5030 CLE$(F)=FICH$(F,3)+FICH$(F,1)
```

CLE$ ( )

BOULOGNE MARTIN
BOULOGNE VINCENT
PARIS DUPONT
PARIS DURAND

Fichier d'adresses

```
10 REM ADR      FICHIER D'ADRESSES
15 GRAB:HIMEM #B3FF                    :REM ou HIMEM #97FF
20 :
25 GOSUB 20000
30 :
50 DIM FICH$(50,4)
60 DIM CLE$(50),IX(50)
70 REM----------
80 :
90 CLS:PRINT "  MODES:":PRINT
100 PRINT SPC(5);"C     : Creation/modification"
105 PRINT SPC(5);"LF    : Liste fichier"
106 PRINT SPC(5);"LFI   : Liste fichier imprimante"
110 PRINT SPC(5);"SAUV  : Sauvegarde casssette"
120 PRINT SPC(5);"LECT  : Lecture cassette"
130 PRINT:INPUT "Mode  ";M$
140 IF M$="C" THEN GOSUB 200
150 IF M$="LECT" THEN GOSUB 480
160 IF M$="SAUV" THEN GOSUB 380
165 IF M$="LF" THEN GOSUB 1000
170 IF M$="LFI" THEN GOSUB 2000
175 IF M$="LFN" THEN GOSUB 4000
176 IF M$="LFV" THEN GOSUB 5000
177 IF M$="S" THEN GOSUB 5230
180 GOTO 80
190 REM================== CREATION/MODICATION ==================
===
200 PRINT:PRINT "Nom?  ";:GOSUB 3000:IF LIG$="" THEN RETURN
210 X$=LIG$:L=LEN(X$)
220 :
230 FOR RANG=1 TO 50
235 :IF FICH$(RANG,1)="" THEN 280
236 :IF FICH$(RANG,1)="*" THEN 280
240 :IF X$=LEFT$(FICH$(RANG,1),L) THEN 330
250 NEXT RANG
255 STOP
260 :
270 REM ---- Nouveau nom
280 :
290 PRINT: PRINT "Rangement:";RANG;:INPUT "Nouveau nom (O/N) ";R$
300 IF R$<>"O" THEN 200
310 FICH$(RANG,1)=X$
```

```
320 REM----
325 PRINT
330 PRINT FICH$(RANG,2);
332 X=15-LEN(FICH$(RANG,2)):IF X>0 THEN PRINT SPC(X);
335 PRINT "Rue? ";:GOSUB 3000:IF LIG$<>"" THEN FICH$(RANG,2)=LIG$
336 :
340 PRINT FICH$(RANG,3);
342 X=15-LEN(FICH$(RANG,3)):IF X>0 THEN PRINT SPC(X);
345 PRINT "Ville? ";:GOSUB 3000:IF LIG$<>"" THEN FICH$(RANG,3)=LIG$
346 :
350 PRINT FICH$(RANG,4);
352 X=15-LEN(FICH$(RANG,4)):IF X>0 THEN PRINT SPC(X);
354 PRINT "Cpost? ";:GOSUB 3000:IF LIG$<>"" THEN FICH$(RANG,4)=LIG$

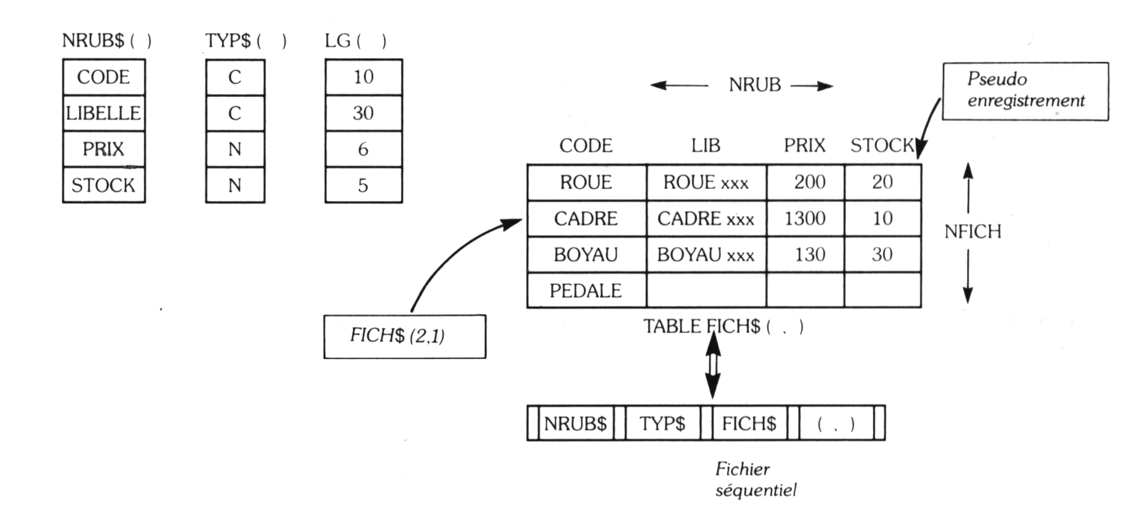
360 GOTO 200
370 REM===================== SAUVEGARDE ======================
380 PRINT:PRINT "APPUYER SUR <RECORD> PUIS <RETURN ":GET X$
390 :
400 PRINT "SAUVEGARDE CASSETTE":WAIT 300
410 POKE #67,0:CALL 1024,FICH$
420 :
460 RETURN
461 :
462 :
463 :
464 :
470 REM===================== LECTURE
480 PRINT:PRINT "APPUYER SUR <PLAY> PUIS <RETURN> ":GET X$
485 PRINT :PRINT "LECTURE CASSETTE":PRINT
490 :
500 POKE #67,0
510 CALL 1027,FICH$
520 :
530 :
560 RETURN
990 REM====================== LISTE FICHIER ====
1000 CLS
1010 PRINT:PRINT "LISTE DU FICHIER:":PRINT
1020 :
1030 FOR F=1 TO 1000
1040 :IF FICH$(F,1)="" THEN 1200
1050 :PRINT FICH$(F,1);
1060 :X=13-LEN(FICH$(F,1)):IF X>0 THEN PRINT SPC(X);
1070 :PRINT FICH$(F,2);
1180 :PRINT
1190 NEXT F
1200 PRINT:PRINT "APPUYEZ SUR <RETURN>":GET X$
1210 RETURN
```

```
1990 REM====================== LISTE FICHIER IMPRIMANTE ==
2000 :
2010 LPRINT:LPRINT "LISTE DU FICHIER":LPRINT
2020 :
2030 FOR F=1 TO 1000
2040 :IF FICH$(F,1)="" THEN 2200
2050 :LPRINT FICH$(F,1);
2060 :LPRINT TAB(13+15);
2070 :LPRINT FICH$(F,2);
2080 :LPRINT TAB(13+35);
2090 :LPRINT FICH$(F,3);
2100 :LPRINT TAB(13+55);
2110 :LPRINT FICH$(F,4);
2180 :LPRINT
2190 NEXT F
2200 :
2210 RETURN
2990 REM------------ INPUT AVEC KEY$ ---
3000 LIG$=""
3010 :
3040 C$=KEY$:IF C$="" THEN 3040
3050 C=ASC(C$):L=LEN(LIG$)
3060 IF C=13 THEN PRINT:RETURN
3065 IF C<>127 THEN 3100
3070 :
3080 IF L>0 THEN LIG$=LEFT$(LIG$,L-1):PRINT CHR$(127);:GOTO 3040 EL
SE 3040
3090 :
3100 IF C<32 THEN PING:GOTO 3040
3110 LIG$=LIG$+C$:PRINT C$;
3120 GOTO 3040
3130 REM-------
3990 REM============ LISTE FICHIER TRIEE PAR NOM ===
4000 :
4010 FOR F=1 TO 100
4020 :IF FICH$(F,1)="" THEN NF=F-1:GOTO 4100
4030 :CLE$(F)=FICH$(F,1):IX(F)=F
4040 NEXT F
4050 STOP
4060 :
4100 GOSUB 4500          ' Appel tri
4105 :
4110 :LPRINT:LPRINT "LISTE TRIEE PAR NOM:":LPRINT
4120 FOR F=1 TO NF
4130 :X=IX(F)
4140 :LPRINT FICH$(X,1);TAB(13+15);
4150 :LPRINT FICH$(X,2);TAB(13+35);
4190 :LPRINT
4200 NEXT F
4210 RETURN
```

```
4490 REM------TRI SHELL
4500 ECART=NF
4505 :
4510 ECART=INT(ECART/2):IF ECART<1 THEN RETURN
4515 :
4520 IV=0
4530 FOR I=1 TO NF-ECART
4540 :J=I+ECART
4550 :IF CLE$(J)=>CLE$(I) THEN 4560
4555 :  X$=CLE$(I):CLE$(I)=CLE$(J):CLE$(J)=X$:X=IX(I):IX(I)=IX(J):IX
(J)=X:IV=1
4560 NEXT I
4570 IF IV=1 THEN 4520 ELSE 4510
4600 REM--------
4990 REM=========== LISTE FICHIER TRIEE PAR VILLE ====
5000 :
5010 FOR F=1 TO 100
5020 :IF FICH$(F,1)="" THEN NF=F-1:GOTO 5100
5030 :CLE$(F)=FICH$(F,3):IX(F)=F
5040 NEXT F
5050 STOP
5060 :
5100 GOSUB 4500          ' Appel tri
5105 :
5110 :LPRINT:LPRINT "LISTE TRIEE PAR VILLE":LPRINT
5120 FOR F=1 TO NF
5130 :X=IX(F)
5140 :LPRINT FICH$(X,3);TAB(13+15);
5150 :LPRINT FICH$(X,1);TAB(13+35);
5190 :LPRINT
5200 NEXT F
5210 RETURN
5220 REM================================ SUPPRESSION ==========
5230 :
5240 PRINT:PRINT "Nom? ";:GOSUB 3000:IF LIG$="" THEN RETURN
5250 NOM$=LIG$
5260 FOR RANG=1 TO 50
5270 :IF FICH$(RANG,1)=NOM$ THEN 5400
5280 NEXT RANG
5290 :PRINT:PRINT "N'EXISTE PAS":PRINT:GOTO 5240
5300 :
5400 INPUT "Suppression OK (O/N) ";R$
5410 IF R$="O" THEN FICH$(RANG,1)="*"
5420 GOTO 5240
```

```
19990 REM-------------------- ROUTINE SAUVEGARDE/LECTURE TABLE --
19995 REM   Cf revue ORIC,septembre 1983 ou Programme ORIC BASE
19996 :
20000 CLS:PRINT "ATTENDEZ":PRINT
20005 A=#B800:READ D$
20010 FOR I=1 TO LEN(D$) STEP 2
20020 : V=VAL("#"+MID$(D$,I,2)):POKE A,V:A=A+1:NEXT I
20030 READ D$:IF D$<>"Z" THEN 20010
20040 DOKE #400,#0A4C:DOKE #402,#4CB8:DOKE #404,#B858
20045 RETURN
20050 DATA 5555555233944363855200BB90820D6B820BAE6A92520C6E5A53320
C6E5A53420
20060 DATA C6E520EEB820A7E5242810032035B82004E82860A000B101F017AAA0
02B10199D0
20070 DATA 0088D0F8E8CAF008B1D120C6E5C8D0F520C3B890DE602095D5200BB9
0820D6B820
20080 DATA 96E62030E6C925D0F92030E685332030E68534A002B1CEC533C8B1CE
E534B00620
20090 DATA 04E84C83C420EEB820EBE4242810032098B82004E82860A000B101F0
1C20F0D4AA
20100 DATA E8A000CAF0082030E691D1C8D0F5A002B9D00091018BD0F820C3B890
D96018A903
20110 DATA65018501A89002E602A502C461E5626020CAE62018B9A003B1CEAA88B
1CEE901B0
20120 DATA01CA85338634601 8A5CE65338561A5CF65348562A004B1CE20F6D1855
F84608501
20130 DATA84026020E800C92CF0034CE4CF4CE200A20020E80086278 5B420E8002
086D1B006
20140 DATA2004E84CE4CFA20086288629 20E20090052086D1900BAA20E20090FB2
086D1B0F6
20150 DATAC924D006A9FF8528D00CC925D00FA980852905B485B48A0980AA20E20
086B5A69E
20160 DATAA59F86CE85CFC5A1D004E4A0F01FA000B1CEC8C5B4D006A5B5D1CEF00
EC8B1CE18
20170 DATA65CEAAC8B1CE65CF90D738602004E8A22A4C85C455
20180 DATA Z
```

GESTION D'UN PSEUDO FICHIER EN MÉMOIRE CENTRALE



Ce programme permet de définir en mémoire centrale un pseudo fichier sauvegardé dans un fichier séquentiel.
La définition des rubriques, faite de façon conversationnelle, est sauvegardée avec le fichier. Créations, modifications et suppressions sont gérées automatiquement.

Liste triée :

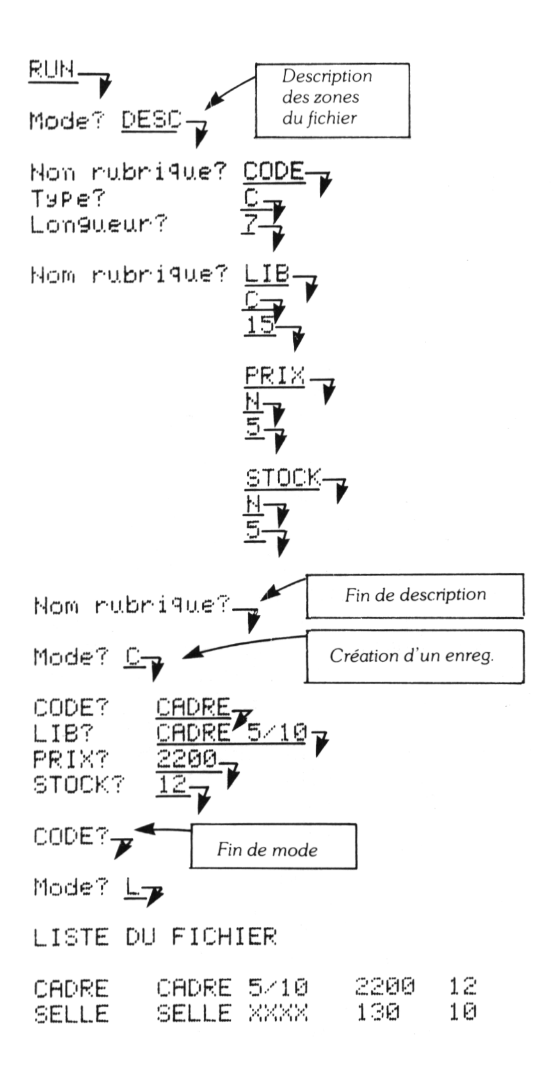
Pour obtenir une liste triée, il suffit d'écrire :

```
ID$(1)="CODE":ID$(2)="PRIX"      ' Noms des zones a editer
PTRI=1                           ' Position tri
GOSUB xxx
```

Des programmes d'application peuvent être ajoutés à la fin du programme standard.

- **Le mode 'VAL' donne le total des stocks**
- **Le mode 'TS' trie le fichier dans l'ordre des stocks**

Remarque : Pour le sous-programme de sauvegarde cassette (20000), voir le programme précédent.



Gestion de fichier

```
10 REM FMEM  GESTION DE FICHIER
15 :
16 GRAB:HIMEM #B3FF              :REM ou HIMEM #97FF
20 GOSUB 20000
21 :
25 DIM NRUB$(10),TYP$(10),LG(13)
26 :
30 :
40 :
50 NFICH=0
55 LG(11)=NFICH
60 :
70 REM  UNE TABLE A 2 DIMENSIONS FICH$(NFICH,NRUB) CONTIENT
80 REM  DANS CHAQUE 'LIGNE' UNE 'FICHE'
90 REM  CETTE TABLE EST SAUVEGARDEE SUE CASSETTE
100 :
110 REM     NFICH    : NOMBRE DE FICHES
120 REM     NRUB     : NOMBRE DE RUBRIQUES
130 REM     MRUB     : MAXIMUM DE FICHES
140 REM     NRUB$( ) : NOMS DES RUBRIQUES
170 REM========================= MENU ===================
190 CLS: PRINT "MODES:":PRINT
200 PRINT SPC(5);"DESC : DESCRIPTION FICHIER"
210 PRINT SPC(5);"VDESC: VISUALISATION DESCRIPTION"
212 PRINT SPC(5);"C     : CREATION/MODIFICATION"
214 PRINT SPC(5);"L     : LISTE FICHIER"
216 PRINT SPC(5);"SAUV : SAUVEGARDE FICHIER"
220 PRINT SPC(5);"LECT : LECTURE FICHIER"
225 PRINT SPC(5);"S     : SUPPRESSION FICHE"
230 PRINT:INPUT "MODE ";M$
240 IF M$="DESC" THEN GOSUB 370
250 IF M$="VDESC" THEN GOSUB 580
260 IF M$="C" THEN GOSUB 700
270 IF M$="SAUV" THEN GOSUB 1130
280 IF M$="LECT" THEN GOSUB 1260
290 IF M$="L" THEN GOSUB 1440
300 IF M$="LT" THEN GOSUB 1570
310 IF M$="S" THEN GOSUB 990
320 IF M$="VAL" THEN GOSUB 2190
330 IF M$="TS" THEN GOSUB 2270
340 GOTO 190
```

```
350 REM======================== DESCRIPTION RUBRIQUES ====
360 :
370 PRINT
380 INPUT "NOMBRE MAXI DE FICHES ";MFICH
385 DIM CTRI$(MFICH),IX%(MFICH):FD=1
390 PRINT
400 FOR I=1 TO 10
410 :PRINT:PRINT "Nom rubrique:";:GOSUB 3000
415 :IF LIG$="" THEN NRUB=I-1:DIM FICH$(MFICH,NRUB):RETURN
420 : NRUB$(I)=LIG$
430 : INPUT "Type rub (C:chaine/N:numerique) ";X$
440 : IF X$<>"C" AND X$<>"N" THEN 430
450 : TYP$(I)=X$
460 : INPUT "Longueur ";X
470 : LG(I)=X
480 NEXT I
490 PRINT
495 PRINT "APPUYEZ SUR <RETURN> ":GET X$
500 RETURN
510 :
520 :
530 :
557 :
558 :
559 :
560 REM=================== VISUALISATION DESCRIPTION ===
570 :
580 PRINT:PRINT "DESCRIPTION DES RUBRIQUES":PRINT
590 FOR I=1 TO 10
600 : IF NRUB$(I)<>"" THEN PRINT NRUB$(I),TYP$(I),LG(I)
610 NEXT I
615 PRINT:PRINT"APPUYER SUR <RETURN>":GET X$
620 PRINT:RETURN
640 :
650 :
690 REM===================== CREATION/MODIFICATION ======
700 :
702 PRINT:PRINT "En mode modification,appuyer sur <RETURN> "
703 PRINT "si la zone ne change pas":PRINT
704 PRINT "R: retour zone arriere":PRINT
705 :
710 PRINT:PRINT "Nom? ";:GOSUB 3000:IF LIG$="" THEN RETURN
720 X$=LIG$:L=LEN(X$)
730 :
740 FOR I=1 TO NFICH
745 :IF FICH$(I,1)="" THEN 760
750 :IF X$=LEFT$(FICH$(I,1),L) THEN RANG=I:GOTO 850
760 NEXT I
765 :
770 PRINT
780 INPUT "NOUVELLE FICHE (O/N) ";R$
785 IF R$<>"O" THEN 710
790 FOR I=1 TO NFICH
800 :IF FICH$(I,1)="*" THEN RANG=I:GOTO 840
810 NEXT I
820 NFICH=NFICH+1:RANG=NFICH
840 FICH$(RANG,1)=X$
```

```
845 REM-----
850 GOSUB 4000           :REM OU GOSUB 870
860 GOTO 710
870 REM--------- SAISIE
880 PRINT
890 FOR I=2 TO NRUB
900 :PRINT NRUB$(I);SPC(8-LEN(NRUB$(I)));FICH$(RANG,I);
905 :PRINT SPC(15-LEN(FICH$(RANG,I)));
910 :GOSUB 3000:IF LIG$="" THEN 940
920 :IF LIG$="R" THEN IF I>2 THEN I=I-1:GOTO 900
930 : FICH$(RANG,I)=LIG$
940 NEXT I
950 RETURN
951 :
955 :
956 :
970 REM================= SUPPRESSION =============
980 :
990 PRINT:INPUT "NOM (F:fin) ";X$:IF X$="F" THEN RETURN
1000 L=LEN(X$)
1010 :
1020 FOR I=1 TO NFICH
1030 :IF X$=LEFT$(FICH$(I,1),L) THEN RANG=I:GOTO 1060
1040 NEXT I
1050 PRINT:PRINT "N'EXISTE PAS":PRINT:GOTO 990
1060 INPUT "ANNULE OK (O/N) ";R$:IF R$<>"O" THEN 990
1070 :
1080 FICH$(RANG,1)="*"
1090 GOTO 990
1091 :
1095 :
1096 REM============= SAUVEGARDE FICHIER ============
1130 PRINT  "APPUYER SUR <RECORD> PUIS <RETURN> "
1140 GET X$
1150 PRINT "SAUVEGARDE CASSETTE":PRINT:WAIT 200
1160 POKE #67,0
1162 :
1164 LG(11)=NFICH:LG(12)=NRUB:LG(13)=MFICH
1165 CALL 1024,LG
1166 PRINT "LG":WAIT 400
1167 :
1170 CALL 1024,NRUB$
1171 WAIT 300
1172 CALL 1024,TYP$
1173 WAIT 300
1175 CALL 1024,FICH$
1176 :
1180 PRINT:PRINT "SAUVEGARDE TERMINEE":PRINT
1190 RETURN
```

```
1260 REM========================= LECTURE FICHIER =========
1270 PRINT:PRINT "APPUYER SUR <PLAY PUIS <RETURN>
1280 GET X$
1290 PRINT "LECTURE CASSETTE":PRINT
1300 POKE #67,0
1305 CALL 1027,LG
1306 PRINT"LG":NFICH=LG(11):NRUB=LG(12):MFICH=LG(13):PRINT LG(11),L
G(12),LG(13)
1307 IF FD=0 THEN DIM FICH$(MFICH,NRUB),CTRI$(MFICH),IX%(MFICH):FD=
1
1308 :
1310 CALL 1027,NRUB$
1312 PRINT NRUB$(1)
1320 CALL 1027,TYP$
1335 CALL 1027,FICH$
1340 PRINT:PRINT "LECTURE TERMINEE":PRINT
1350 RETURN
1420 REM===================== LISTE DES FICHES =========
1430 :
1440 ID(1)=1:ID(2)=2
1450 GOTO 1480
1460 RETURN
1470 :
1480 PRINT:PRINT "LISTE DES FICHES":PRINT
1490 FOR I=1 TO NFICH:IX%(I)=I:NEXT I
1500 GOSUB 1830
1510 RETURN
1550 REM=================== LISTE TRIEE DES FICHES ======
1560 :
1570 ID$(1)="CODE":ID$(2)="PRIX":ID$(3)="STOCK"
1575 :
1580 PTRI=1         ' Position tri
1590 GOSUB 1620
1600 RETURN
1610 REM------
1620 FOR I=1 TO 10
1630 :FOR R=1 TO NRUB
1640 : IF ID$(I)=NRUB$(R) THEN ID(I)=R
1650 :NEXT R
1660 NEXT I
1670 :
1680 FOR I=1 TO NFICH
1682 :CTRI$(I)=FICH$(I,PTRI):IX%(I)=I
1684 NEXT I
1700 GOSUB 1990
1702 :
1703 :
1704 :
1710 PRINT:PRINT "LISTE TRIEE DES FICHES":PRINT
1720 GOSUB 1830
1730 RETURN
```

```
1740 REM-- SPGM EDITION ECRAN
1830 :
1860 : FOR F=1 TO NFICH
1870 :
1880 : F1=IX%(F)
1885 : PRINT "!";
1890 : FOR R=1 TO NRUB
1900 :  IF ID(R)=0 THEN 1950
1910 :  PRINT LEFT$(FICH$(F1,ID(R)),LG(ID(R)));
1915 :  X=LG(ID(R))-LEN(FICH$(F1,ID(R))):IF X>0 THEN PRINT SPC(X);
1917 :  PRINT "!";
1940 : NEXT R
1950 : PRINT
1960 NEXT F
1970 PRINT:PRINT "APPUYER SUR <RETURN> ":GET X$
1975 RETURN
1980 REM----------------- TRI SHELL-METZNER
1990 PAS=NFICH
2000 PRINT:PRINT "JE TRIE POUR VOUS":PRINT
2010 :
2020 PAS=INT(PAS/2):IF PAS<1 THEN RETURN
2030 J=1:K=NFICH-PAS
2040 :
2050 I=J
2060 :
2070 L=I+PAS
2080 IF CTRI$(I)<CTRI$(L) THEN 2120
2090 X$=CTRI$(L):CTRI$(L)=CTRI$(I):CTRI$(I)=X$
2095 X=IX%(L):IX%(L)=IX%(I):IX%(I)=X
2100 I=I-PAS:IF I<1 THEN 2120
2105 GOTO 2070
2110 :
2120 J=J+1:IF J>K THEN 2020
2125 GOTO 2050
2170 REM============ VALORISATION STOCK =============
2180 :
2190 TT=0
2200 FOR F=1 TO NFICH
2210 :TT=TT+VAL(FICH$(F,3))*VAL(FICH$(F,4))
2220 NEXT F
2230 PRINT:PRINT "TOTAL STOCK:";TT
2235 PRINT:PRINT "APPUYEZ SUR <RETURN> ":GET X$
2240 RETURN
2990 REM============== INPUT AVEC KEY$ ====
3000 LIG$=""
3040 C$=KEY$:IF C$="" THEN 3040
3050 C=ASC(C$):L=LEN(LIG$)
3060 IF C=13 THEN PRINT:RETURN
3065 IF C<>127 THEN 3100
3070 :
3080 IF L>0 THEN LIG$=LEFT$(LIG$,L-1):PRINT CHR$(127);:GOTO 3040 EL
SE 3040
3090 :
3100 IF C<32 THEN PING:GOTO 3040
3110 LIG$=LIG$+C$:PRINT C$;
3120 GOTO 3040
```

SAISIE ÉCRAN AVEC KEY$

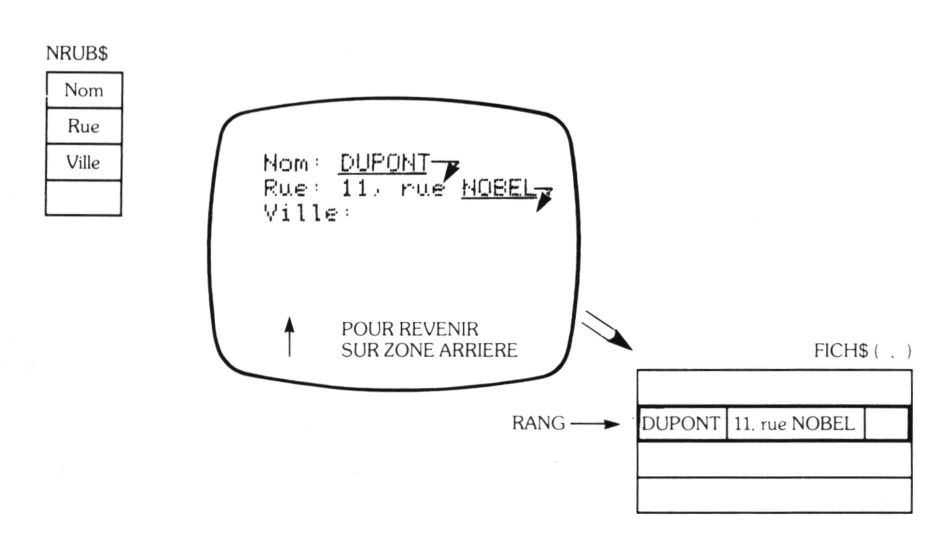
La saisie d'informations à l'aide d'une suite d'instructions **'INPUT'** ne permet pas, en cours de saisie, de se positionner sur une zone arrière qui aurait été mal documentée.

```
10 INPUT "Nom ";NOM$
20 INPUT "Rue ";RUE$
30 INPUT "Ville";VILLE$

RUN

Nom?    DUPONT
Rue?    Rue NAUBEL
Ville? MONTIGNY
```

Nous proposons ici un sous-programme qui permet de modifier des zones mal documentées. Il est utilisable avec le programme précédent. Une table NRUB$ contient les noms des zones à saisir. Le résultat est rangé dans une table FICH$ (RANG,)



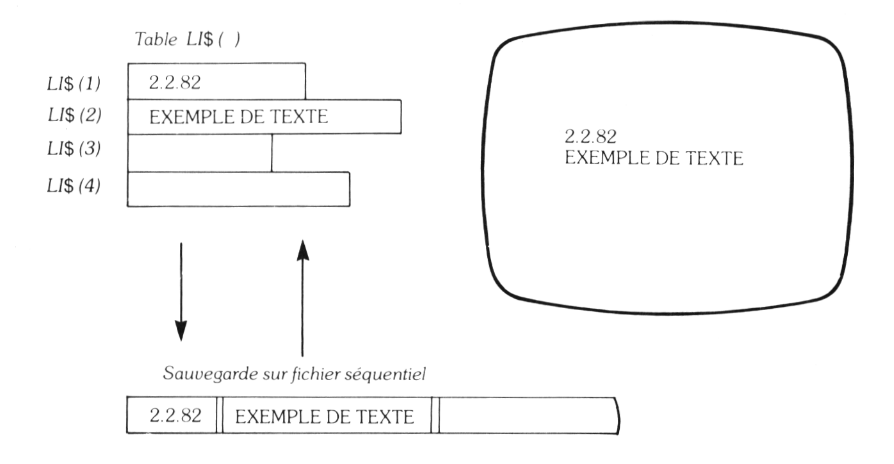
Saisie écran avec KEY$

```
3121 :
3122 :
3123 :
3124 :
3990 REM=================== SAISIE ECRAN NO2 ====
4000 :
4010 CLS
4015 PLOT 2,23,"^ Pour zones arrieres"
4020 FOR L=2  TO NRUB
4030 : PLOT 1,L+1,NRUB$(L)
4035 : PLOT 15,L+1,FICH$(RANG,L)
4040 :
4100 NEXT L
4110 REM------------------ SAISIE DES ZONES --
4120 FOR P=2 TO NRUB
4130 :
4140 :XL=15:YL=P+1
4145 :GOSUB 5000
4160 :IF R=1 THEN FICH$(RANG,P)=LIG$
4170 :IF R=2 THEN 4200
4180 :IF R=3 THEN IF P>1 THEN PLOT XL,YL,FICH$(RANG,P):P=P-1:GOTO 4
140
4200 : PLOT XL,YL ,FICH$(RANG,P)
4205 NEXT P
4210 RETURN
4990 REM------- Saisie d'une ligne
5000 LIG$=""
5010 :
5020 L=LEN(LIG$):PLOT XL+L,YL,132
5030 :
5040 C$=KEY$:IF C$="" THEN 5040
5050 C=ASC(C$):L=LEN(LIG$)
5055 IF C=13 THEN 5140
5060 :
5061 IF C=11 THEN R=3:RETURN
5065 IF C<>127 THEN 5100
5070 :
5080 IF L>0 THEN LIG$=LEFT$(LIG$,L-1):PLOT XL+L,YL," ":GOTO 5020 EL
SE 5040
5090 :
5100 IF C<32 THEN PING:GOTO 5040
5110 LIG$=LIG$+C$:PLOT XL,YL,LIG$
5120 GOTO 5020
5130 :
5140 PLOT XL+L,YL," "
5150 IF LIG$<>"" THEN R=1 ELSE R=2
5160 RETURN
```

ÉDITEUR DE TEXTE

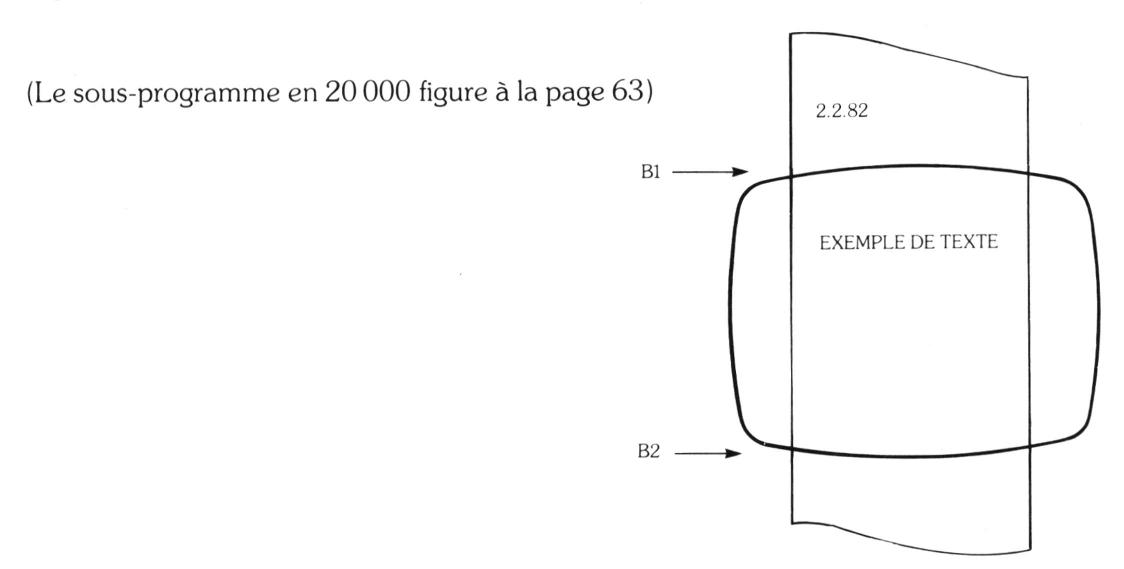
Editeur de texte

Ce programme permet de saisir et de modifier un texte. L'image de l'écran représente le texte.



L'insertion de caractères dans une ligne se fait immédiatement dès la frappe de ceux-ci (il n'y a pas de mode insertion).
Des caractères spéciaux sont prévus pour la suppression de caractères, la suppression et l'insertion de lignes.
22 lignes de texte seulement peuvent être visualisées à la fois. Les caractères '!' et '"' déplacent la partie de texte visualisée vers le haut et vers le bas.

(Le sous-programme en 20 000 figure à la page 63)



Editeur

```
10 REM  EDT   EDITEUR
15 :
20 GOSUB 20000          :REM Pour sauvegarde cassette
25 INK4
26 PRINT CHR$(17);      :REM Suppression curseur
40 CLS:PRINT "EDITEUR PLEIN ECRAN":PRINT
50 PRINT " 4 fleches pour se deplacer"
60 PRINT " DEL pour effacer un caractere"
70 PRINT " CTRL A pour sauvegarde
75 PRINT " CTRL B pour edition imprimante"
80 PRINT " !       decalage haut"
90 PRINT " '       decalage bas "
100 PRINT " @       insertion ligne
135 C$="":C=0:LI$="":L=0:PC=0
140 DIM LI$(50)
150 :
170 PRINT
180 INPUT "Nouveau texte (O/N) ";R$
185 IF R$="O" THEN CLS:GOTO 210
190 GOSUB 2000
191 :
192 CLS
193 FOR LI=1 TO 22
194 : PLOT 1,LI-1,LI$(LI)
196 NEXT LI
200 :
210 :
300 B1=1:B2=22
310 PC=1                 :REM Position curseur ligne courante
320 FOR LI=B1 TO B2
330 :
340 :IF LI>MX THEN MX=LI
350 :LI$=LI$(LI):GOSUB 580
360 :PLOT 1,LI-B1,LI$+" "
370 :LI$(LI)=LI$
380 :IF R=2 THEN IF LI>1 THEN LI=LI-1:GOTO 340
385 :IF R=2 THEN IF LI<=1 THEN 340
390 :IF R=3 THEN IF LI<B2 THEN LI=LI+1:GOTO 340
395 :IF R=3 THEN IF LI=>B2 THEN 340
396 :IF R=10 THEN GOSUB 3100
400 REM------ Ajout 1 ligne
410 IF R<>4 THEN 420
415 :FOR I=MX TO LI STEP-1:LI$(I+1)=LI$(I):NEXT I:LI$(LI)=""
416 :CLS:FOR I=B1 TO B2
417 :PRINT LI$(I)
418 :NEXT I
419 :MX=MX+1:GOTO 340
420 :
440 :REM------ Sauvegarde
450 :IF R=6 THEN CLS:GOSUB 3000:GOTO 192
460 REM-------Decalage bas
470 :IF R<>7 THEN 480
475 :B1=B1+1:B2=B2+1:CLS
476 :FOR I=B1 TO B2:PRINTLI$(I):NEXT I
477 :LI=LI+1:GOTO 340
```

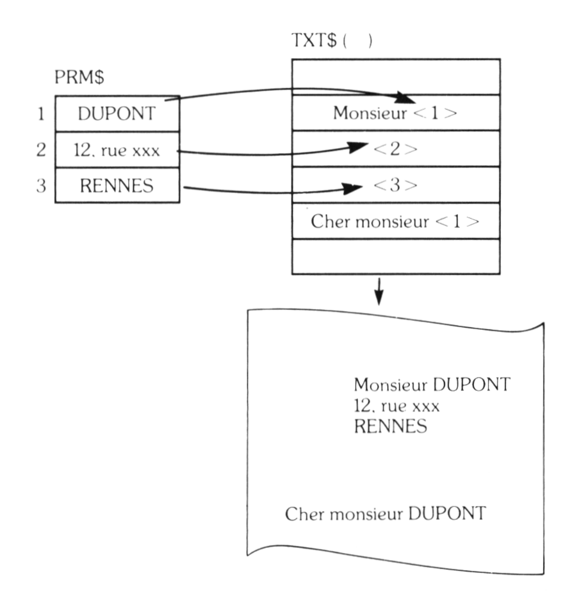
```
480 REM--------Decalage haut
490 :IF R<>8 THEN 500
495 :IF B1<=1 THEN 340
496 :B1=B1-1:B2=B2-1:CLS
497 :FOR I=B1 TO B2:PRINT LI$(I):NEXT I:LI=LI-1:GOTO 340
498 :
500 NEXT LI
501 :
502 :
503 :
504 :
505 :
506 :
570 REM------------------ Saisie d'une ligne
580 :
590 IF PC>LEN(LI$) THEN PC=LEN(LI$)+1
600 IF LI$="" THEN PC=1
610 :
620 PLOT PC,LI-B1,132
630 :
640 C$=KEY$
645 IF C$="" THEN PLOT PC,LI-B1,MID$(LI$,PC,1):GOTO 620
647 :
650 C=ASC(C$)
660 L=LEN(LI$)
662 IF C>127 THEN 640
664 IF C=11 THEN R=2:RETURN
670 IF C=13 THEN R=3:RETURN
680 IF C=8 THEN 930
690 REM---- Effacement caractere
695 IF C<>127 THEN 710
696 IF (L-PC+1)<=0 THEN 640
700 LI$=LEFT$(LI$,PC-1)+RIGHT$(LI$,L-PC):PLOT 1,LI-B1,LI$+" ":GOTO
620
710 REM--- Curseur droite
720 IF C=9 THEN IF PC<=L THEN PLOT PC-0,LI-B1,MID$(LI$,PC,1):PC=PC+
1:GOTO 620
725 IF C=9 THEN IF PC>L THEN 640
730 REM--
760 IF C=10 THEN R=3:RETURN
770 IF C=64 THEN R=4:RETURN :REM @ ajout ligne
790 IF C=1 THEN R=6:RETURN    :REM Sauvegarde
795 IF C=2 THEN R=10:RETURN
800 IF C=33 THEN R=7:RETURN :REM ! bas
810 IF C=39 THEN R=8:RETURN :REM ' haut
815 IF C=3 THEN PRINT CHR$(17):END
820 IF C<32 THEN 640
840 REM-----caractere normal concatene
850 IF L>36 THEN 640
860 LI$=LEFT$(LI$,PC-1)+C$+RIGHT$(LI$,L-PC+1)
870 PLOT 1,LI-B1,LI$
880 :
890 :
900 PC=PC+1
910 GOTO 620
```

```
920 REM---8
930 IF PC<=1 THEN 640
940 IF PC>L THEN PLOT PC,LI-B1," "
950 PLOT PC-0,LI-B1,MID$(LI$,PC,1)
960 PC=PC-1
970 GOTO 620
1990 REM==================== LECTURE CASSETTE ===
2000 PRINT:PRINT "APPUYER SUR <PLAY> PUIS <RETURN>":GET X$
2010 PRINT:PRINT "LECTURE CASSETTE":PRINT
2020 POKE #67,0
2030 CALL 1027,LI$
2040 :
2050 RETURN
2060 :
2061 :
2062 :
2063 :
2064 :
2065 :
2066 :
2067 :
2990 REM==================== SAUVEGARDE CASSETTE ======
3000 PRINT:PRINT "APPUYER SUR <RECORD> PUIS <RETURN> ":GET X$
3010 :
3020 PRINT :PRINT "SAUVEGARDE CASSETTE":PRINT
3030 :
3040 POKE #67,0
3045 WAIT 200
3050 CALL 1024,LI$
3060 RETURN
3090 REM=============== EDITION IMPRIMANTE ======
3100 LPRINT
3110 FOR L=1 TO MX
3120 :LPRINT LI$(L)
3130 NEXT L
3140 RETURN
```

```
 ***   ***   ***   ***
*   *  *  *   *   *
*   *  ***    *   *
*   *  *  *   *   *
 ***   *  *  ***   ***
```

COURRIER PARAMÉTRÉ

Une lettre standard est définie en 'DATA', ainsi qu'une liste de destinataires.
Le programme personnalise cette lettre pour chacun des destinataires.
Nous lisons d'abord la lettre définie en 'DATA' dans une table TXT$() puis pour chaque ligne de la lettre, nous remplaçons les paramètres entre <> par les valeurs réelles de chacun des destinataires.



Courrier paramétré

```
10 REM COURP   COURRIER PARAMETRE
20 :
30 DIM TXT$(30)
35 REM------------------------- Lettre ---------
40 DATA "LIBERT Pierre                            Le 28.8.83"
50 DATA "Rue de PARIS
60 DATA "61-SEPT FORGES"
70 DATA "   "
80 DATA "   "
90 DATA "   "
100 DATA "         Monsieur <1>   "
110 DATA "   "
120 DATA "          <2>           "
130 DATA "          <3>           "
140 DATA "   "
150 DATA " Cher Monsieur <1> ,................    "
160 DATA " .................................    "
170 DATA "   "
180 DATA "   "
190 DATA "                         LIBERT P."
200 DATA "   "
210 DATA "   "
290 DATA *
300 REM--------------- Adresses envois ---------
310 DATA DUPONT,12 RUE XXX,RENNES
320 DATA FORTIN,RUE YYYY,RENNES
330 DATA BETHUEL,RUE ZZZZ,PARIS
390 DATA *,*,*
400 REM------------ Lecture lettre dans TXT$()
410 :
420 FOR L=1 TO 100
430 :READ TXT$(L):IF TXT$(L)="*" THEN NL=L-1:GOTO 500
440 NEXT L
490 REM---------------------- Edition
500 FOR EV=1 TO 100
510 :FOR P=1 TO 3:READ PRM$(P):NEXT P
520 :IF PRM$(1)="*" THEN STOP
530 :
540 :FOR LG=1 TO NL
550 :  LIGNE$=TXT$(LG)
560 :  CCH$="<":GOSUB 3000:P1=P
565 :  IF P1=0 THEN 600
570 :  NP=VAL(RIGHT$(LIGNE$,LEN(LIGNE$)-P1 ))
575 :  CCH$=">":GOSUB 3000:P2=P
580 :
585 :  LIGNE$=LEFT$(LIGNE$,P1-1)+PRM$(NP)+RIGHT$(LIGNE$,LEN(LIGNE$)
-P2)
590 :
600 :  LPRINT LIGNE$
620 :NEXT LG
640 :PRINT:PRINT "Appuyez sur <RETURN> ":GET X$
650 NEXT EV
660 :
700 STOP
```

```
2990 REM-------------Recherche sous chaine
3000 L1=LEN(CCH$)
3010 :
3020 FOR P=1 TO LEN(LIGNE$)-L1
3030 : IF MID$(LIGNE$,P,L1)=CCH$ THEN 3100
3040 NEXT P
3050 P=0
3100 RETURN
```

```
LIBERT Pierre                     Le 28.8.83
Rue de PARIS
61-SEPT FORGES



         Monsieur DUPONT

          12 RUE XXX
          RENNES

 Cher Monsieur DUPONT ,................
 ..................................


                       LIBERT P.
```

```
LIBERT Pierre                     Le 28.8.83
Rue e PARIS
61-SEPT FORGES



         Monsieur FORTIN

          RUE YYYY
          RENNES

 Cher Monsieur FORTIN ,................
 ..................................


                       LIBERT P.
```

Quatrième partie
JEUX

JEU DES ALLUMETTES

Règle du jeu

Au départ, il y a 30 allumettes.

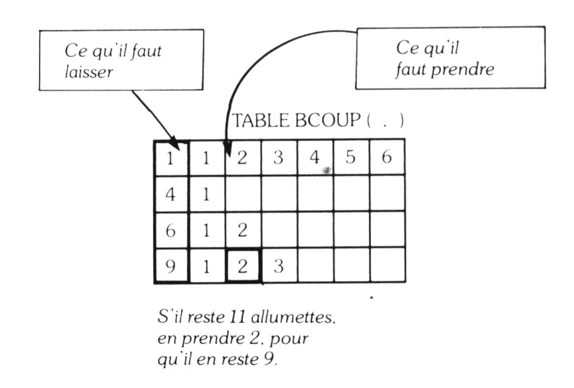
- Les 2 joueurs prennent des allumettes à tour de rôle. Celui qui prend la dernière perd.
- Chaque joueur peut prendre entre une et le double des allumettes prises par le joueur précédent. Par exemple, si le joueur A prend 2 allumettes, le joueur B peut en prendre 1, 2, 3 ou 4.

```
RUN

Voulez vous commencer? (O/N) O
Combien?        1
J'en Prends     2

Il en reste: 27
Combien?        3
     :
```

Nous avons défini une table BCOUP(,) où est stockée dans la 1re colonne le jeu qu'il faut à l'adversaire et dans les autres colonnes le nombre d'allumettes qu'il faut prendre pour gagner. Par exemple, s'il reste 11 allumettes, prenez en 2 afin qu'il en reste 9.



1	1	2	3	4	5	6
4	1					
6	1	2				
9	1	2	3			

S'il reste 11 allumettes, en prendre 2, pour qu'il en reste 9.

Jeu des allumettes

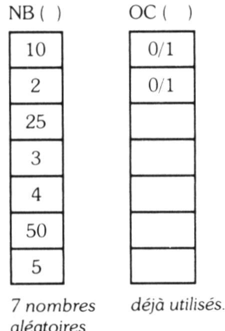
```
10 REM ALU  JEU DES ALLUMETTES
20 :
30 :
100 DATA 01,1,2,3,4,5,6
110 DATA 04,1,.,.,.,.,.
120 DATA 06,1,2,.,.,.,.
130 DATA 09,1,2,3,.,.,.
140 DATA 12,1,.,.,.,.,.
150 DATA 14,1,2,3,4,5,6
160 DATA 17,1,.,.,.,.,.
170 DATA 19,1,2,.,.,.,.
180 DATA 22,1,2,3,4,.,.
190 DATA 25,1,.,.,.,.,.
200 DATA 27,1,2,.,.,.,.
210 :
220 DIM BCOUP(13,7):FOR J=1 TO11:FOR I=1 TO 7:READ BCOUP(J,I):NEXT
I:NEXT J
230 DEBUT=30
240 :
250 CLS:PRINT "  JEU DES ALLUMETTES":PRINT
260 PRINT "AU DEPART,IL Y A 30 ALLUMETTES"
265 PRINT "CHACUN PREND DES ALLUMETTES
267 PRINT "ALTERNATIVEMENT."
269 PRINT
270 PRINT "CELUI QUI PREND LA DERNIERE PERD":PRINT
280 PRINT "ON PEUT EN PRENDRE ENTRE 1 ET LE"
285 PRINT "DOUBLE DU JEU PRECEDENT"
290 :
300 NRESTE=DEBUT:NMX=2:NCOUP=1:PFINI=0
310 PRINT:INPUT "Voulez vous commencer(O/N)";R$:IF R$="O" THEN TR=1
 ELSE TR=2
320 :
330 ON TR GOSUB 370,480
340 IF PFINI=1 THEN GOSUB 570:GOTO 300
350 GOTO 330
360 REM--------------- JEU JOUEUR
370 PRINT:INPUT "Combien en Prenez vous ";NPRISE
380 IF NPRISE<1 THEN 370
390 IF NPRISE>NMX THEN PRINT: PRINT NMX;" AU PLUS":GOTO 370
400 IF NPRISE>NRESTE THEN PRINT:PRINT "IL N'EN RESTE QUE ";NRESTE:G
OTO 370
410 :
420 IF NPRISE=NRESTE THEN PRINT :PRINT "J'AI GAGNE ":PFINI=1:RETURN
430 NRESTE=NRESTE-NPRISE:NMX=NPRISE*2
440 TR=2
450 RETURN
```

```
460 REM---------------- JEU MACHINE
470 :
480 IF NRESTE=1 THEN PRINT "J'AI PERDU":PFINI=1:RETURN
490 :
500 IF NMX=>NRESTE THEN NPRISE=NRESTE-1:GOTO 510 ELSE GOSUB 610
510 PRINT:PRINT "J'EN PRENDS: ";NPRISE
520 NRESTE =NRESTE-NPRISE:NMX=NPRISE*2
530 PRINT :PRINT "IL EN RESTE:";NRESTE
540 IF NRESTE=1 THEN PRINT:PRINT "J'AI GAGNE":PRINT:PFINI=1:RETURN
550 TR=1
560 RETURN
561 :
562 :
563 :
565 :
567 :
568 :
569 :
570 PRINT:INPUT "COMBIEN D'ALLUMETTES POUR LA PARTIE ";DEBUT
580 GOTO 300
600 REM---------- Recherche dans la table des bons coups
610 FOR L=11 TO 1 STEP-1
620 : FOR C=2 TO 6
630 :   IF BCOUP(L,C)=0 THEN 660
640 :   IF BCOUP(L,1)+BCOUP(L,C)<>NRESTE THEN 650
645 :   IF BCOUP(L,C)>NMX THEN 650
647 :   NPRISE=BCOUP(L,C):RETURN
650 : NEXT C
660 NEXT L
670 NPRISE=1:PRINT:PRINT "VOUS POURRIEZ GAGNER":RETURN
```

LE COMPTE EST BON

Nous proposons une version simplifiée du jeu **'le compte est bon' :**

- Nous choisissons 7 nombres aléatoires et nous effectuons des opérations aléatoires avec ces nombres. Puis nous affichons les 7 nombres aléatoires ainsi que le résultat calculé.
- Ce programme ne vérifie pas si le résultat calculé par le joueur est exact. Il faudrait pour cela 'évaluer' l'expression donnée par le joueur (par exemple : (25 – 3) * (5 – 2))



Les opérations générées par le programme sont seulement du type :

Résultat = Résultat précédent { + – * / } **nombre aléatoire**

```
RUN
Voici les 7 nombres:
10 2 25 3 4 50 5
Resultat atrouver: 66
Appuyer sur <RETURN>
Ma solution: ((25+3)*2)+10)
```

AMÉLIORATIONS POSSIBLES

- N'afficher que les parenthèses strictement nécessaires, par exemple (25 + 3) * 2 + 10.
- Pour évaluer le résultat donné par le joueur, voir le programme 'évaluation d'une expression'.

Le compte est bon

```
10 REM CPTB1 LE COMPTE EST BON NO1
20 :
60 CLS
90 REM------------- 7 NOMBRES ALEATOIRES --
100 FOR N=1 TO 7
110 : NB(N)=1+INT(RND(1)*14)
120 : IF NB(N)>10 THEN NB(N)=(NB(N)-10)*25
130 :NEXT N
140 REM------- Nombre d'operations (a adapter)
150 NP=3+INT(RND(1)*3)              ' 3 ->6 operations
152 :
155 FOR I=1 TO 7:OC(I)=0:NEXT I   :REM  Nombres deja utilises
160 :
170 PX=INT(RND(1)*7)+1
175 IF NB(PX)<>1 THEN RESUL=NB(PX):OC(PX)=1
180 :
182 R$="":FOR I=1 TO NP-1:R$=R$+"(":NEXT I
185 R$=R$+STR$(RESUL)
200 FOR N=1 TO NP-1
210 : OP=INT(RND(1)*4)+1            ' Operation
215 :
220 : PX=1+INT(RND(1)*7)            ' Choix operande
230 : IF OC(PX)=1 THEN 220          ' Deja Pris
240 :
250 : ON OP GOTO 300,400,500,600
260 :
300 : RESUL=RESUL+NB(PX):OP$="+"
310 : GOTO 700
320 :
400 : X=RESUL-NB(PX):IF X<1 THEN 210
405 : RESUL=X:OP$="-"
410 : GOTO 700
420 :
500 : IF NB(PX)=1 THEN 220
510 : RESUL=RESUL*NB(PX)
520 : OP$="*"
530 : GOTO 700
540 :
600 : IF NB(PX)=1 THEN 210
605 : X=RESUL/NB(PX)
610 : IF X<>INT(X) THEN 210
620 : RESUL=X:OP$="/"
640 : GOTO 700
650 :
700 : OC(PX)=1
710 : R$=R$+OP$+STR$(NB(PX))+")"
800 NEXT N
810 IF RESUL<100 OR RESUL>999 THEN 150
820 :
830 CLS
840 PRINT "VOICI LES 7 NOMBRES:":PRINT
890 FOR N=1 TO 7:PRINT NB(N);:NEXT N:PRINT
900 :
910 PRINT:PRINT "Resultat a trouver:";RESUL:PRINT
```

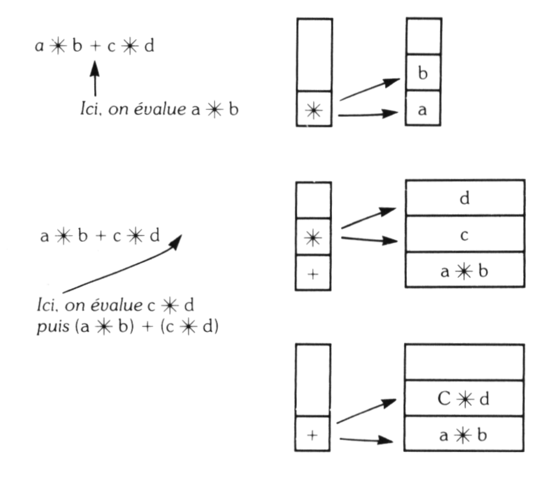
```
930 PRINT "Pour avoir ma solution faire <RETURN>"
940 GET X$
950 PRINT:PRINT R$
960 :
965 PRINT
970 PRINT "APPUYER SUR <ENTER> ":GET X$
980 GOTO 100
```

ÉVALUATION D'EXPRESSIONS

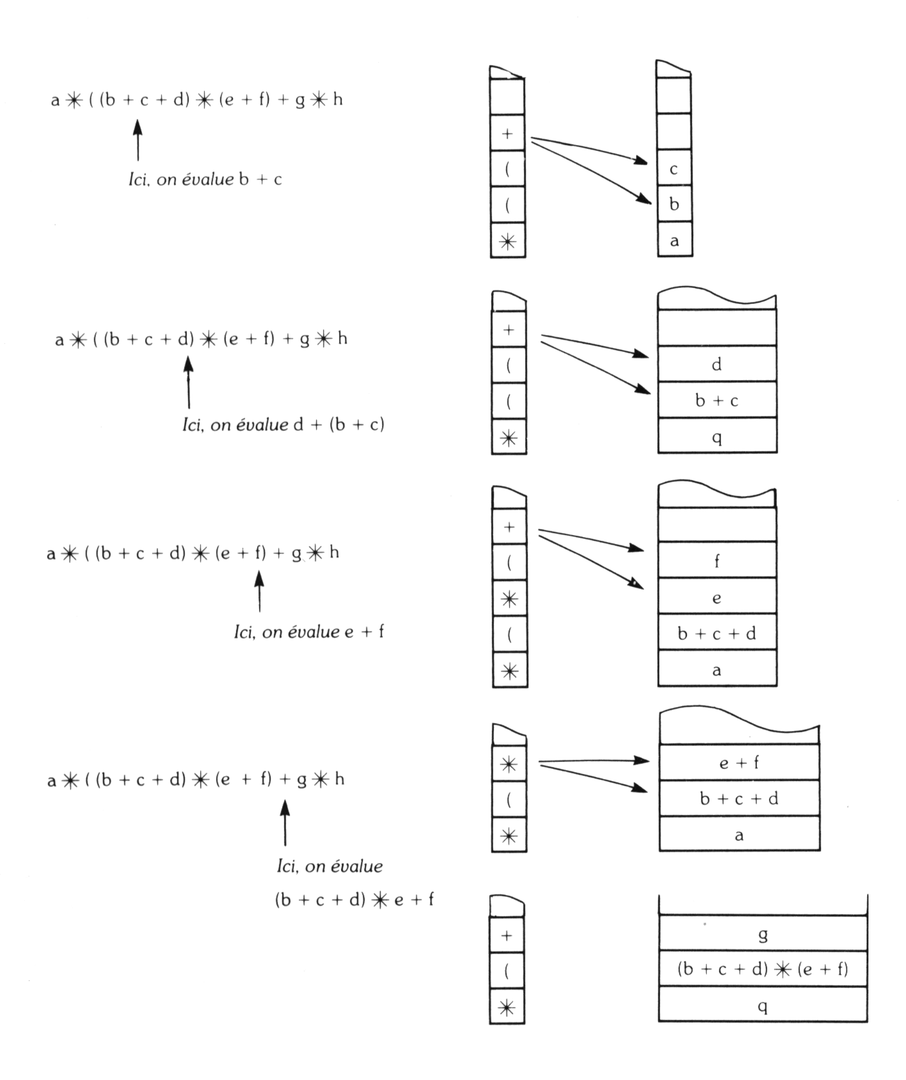
Etudions comment peut être réalisée l'évaluation d'une expression arithmétique.

L'analyse d'une expression se fait de gauche à droite. Chaque opérande rencontré est mis dans une pile dite 'pile des opérandes', chaque opérateur étant lui stocké dans une 'pile des opérateurs'.

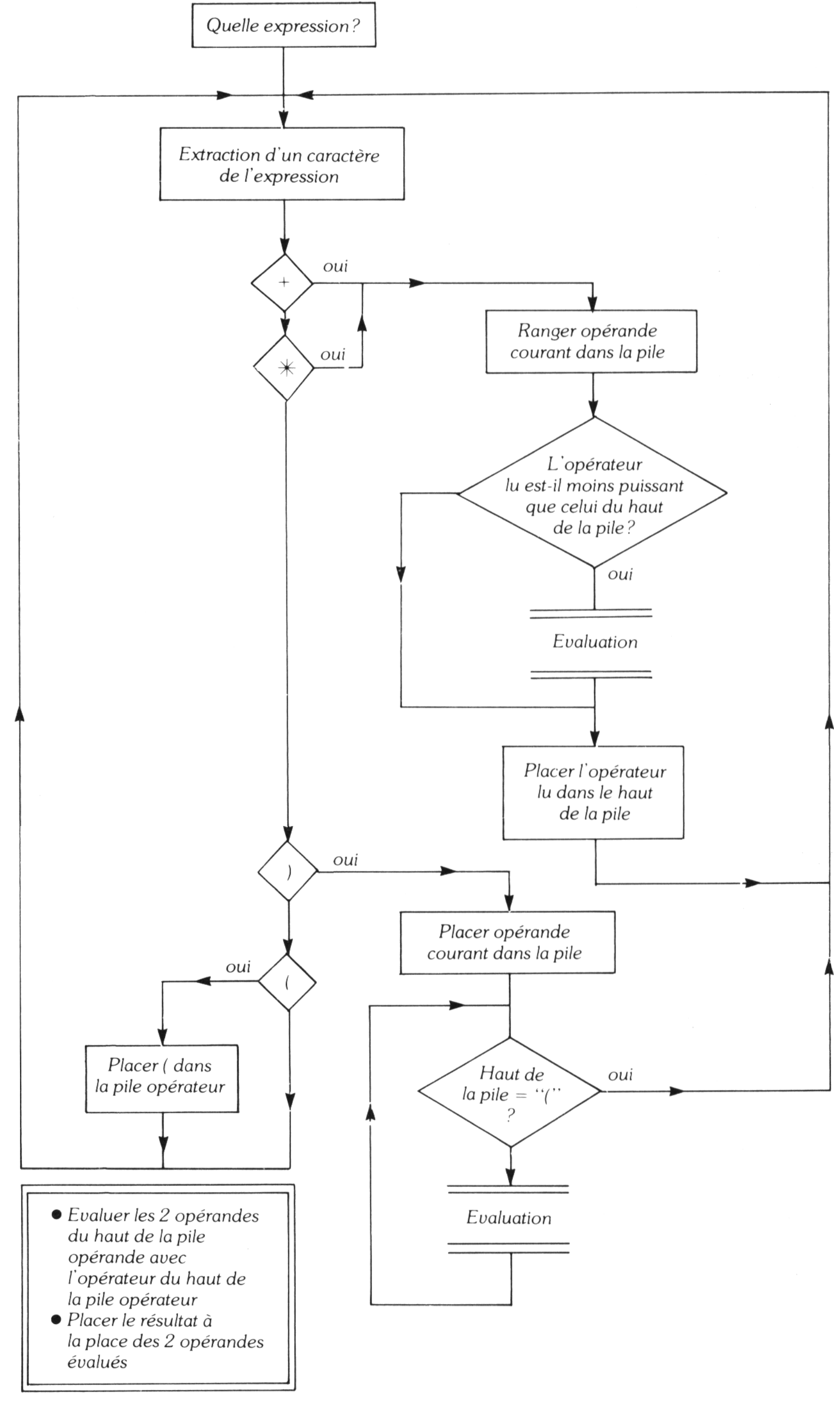
A chaque fois qu'un opérateur de puissance inférieure ou égale à celle de l'opérateur du haut de la pile est rencontré dans l'expression à évaluer (+ est moins puissant que *), on évalue le résultat de l'opération entre les deux opérandes du haut de la pile avec l'opérateur du haut de la pile, puis le résultat est rangé à la place des deux précédents opérandes. Enfin, l'opérateur du haut de la pile est supprimé.



Lorsque l'expression à évaluer comporte des parenthèses, celles ouvrantes sont stockées systématiquement dans la pile des opérateurs. Considérées plus puissantes que les opérateurs, elles empêchent l'évaluation tant qu'une parenthèse fermante n'est pas rencontrée. Chaque parenthèse fermante provoque l'évaluation jusqu'à la parenthèse ouvrante correspondante.



Remarques : Compte tenu des temps de traitement des chaînes de caractères, ce programme est relativement long à exécuter. Nous n'avons considéré que les opérations d'addition et de multiplication, celles dont nous aurons besoin plus loin.



Evaluation d'expressions

```
10 REM EVA       EVALUATION D'UNE EXPRESSION
20 :
40 DATA "*",1,"/",1,"+",2,"-",2,")",3,"(",4
50 :
60 FOR I=1 TO 6:READ P$(I),P(I):NEXT I
90 X$="(8+3)*12/2)"       :REM Remplacer par INPUT X$
100 :
120 RT=1:RD=1
130 CUR=1
140 :
150 DEP=CUR
160 :
170 Y$=MID$(X$,CUR,1)        ' Y$:caractere analyse
180 :
190 IF Y$="" THEN Y$=")": GOSUB 280:PRINT:PRINT "RESULTAT:";RA(1):S
TOP
200 IF Y$="+" THEN GOSUB 280:CUR=CUR+1:GOTO 150
205 IF Y$="-" THEN GOSUB 280:CUR=CUR+1:GOTO 150
210 IF Y$="*" THEN GOSUB 280:CUR=CUR+1:GOTO 150
215 IF Y$="/" THEN GOSUB 280:CUR=CUR+1:GOTO 150
220 IF Y$=")" THEN GOSUB 330:RT=RT-1:CUR=CUR+1:GOTO 150
230 IF Y$="(" THEN RT$(RT)=Y$:RT=RT+1:CUR=CUR+1:GOTO 150
240 CUR=CUR+1
250 GOTO 170
260 REM--------------------- PLUS/MULT
280 GOSUB 440
281 :
282 XX$=Y$:GOSUB 800:P1=P
284 IF RT>1 THEN XX$=RT$(RT-1):GOSUB 800:P2=P
290 IF RT>1 AND P1=>P2 THEN GOSUB 380:RT=RT-1:GOTO 282
295 :
300 RT$(RT)=Y$:RT=RT+1
310 RETURN
320 REM--------------------- PARENTHESE FERMANTE ----
330 GOSUB 440
331 :
332 XX$=Y$:GOSUB 800:P1=P
334 XX$=RT$(RT-1):GOSUB 800:P2=P
340 IF RT>1 AND P1=>P2 THEN GOSUB 380:RT=RT-1:GOTO 332
350 RETURN
360 REM---------------------- EVALUATION -----
380 FOR I=1 TO RT-1:LPRINT RT$(I);:NEXT I
390 LPRINT TAB(25);:FOR I=1 TO RD-1:LPRINT RA(I);:NEXT I:LPRINT
400 IF RT$(RT-1)="*" THEN RA(RD-2)=RA(RD-1)*RA(RD-2):RD=RD-1:RETURN
405 IF RT$(RT-1)="/" THEN RA(RD-2)=RA(RD-2)/RA(RD-1):RD=RD-1:RETURN
410 IF RT$(RT-1)="+" THEN RA(RD-2)=RA(RD-1)+RA(RD-2):RD=RD-1:RETURN
415 IF RT$(RT-1)="-" THEN RA(RD-2)=RA(RD-2)-RA(RD-1):RD=RD-1:RETURN
420 REM------------------ AJOUT PILE OPERANDE ----
440 CH$=MID$(X$,DEP,CUR-DEP)
450 IF CH$="" THEN RETURN
460 RA(RD)=VAL(CH$)
470 RD=RD+1
480 RETURN
```

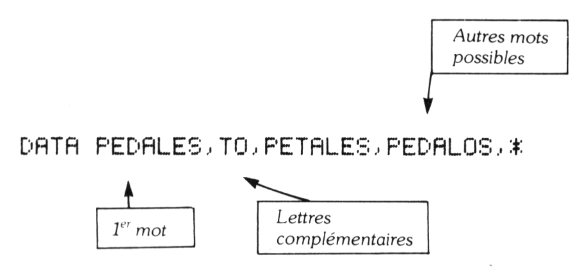
```
790 REM----------- POIDS OPERATEUR ----
800 FOR I=1 TO 6
810 :IF XX$=P$(I) THEN P=P(I):RETURN
820 NEXT I
830 P=0:RETURN
```

```
(+          8 3
*           11 12
/           132 2
```

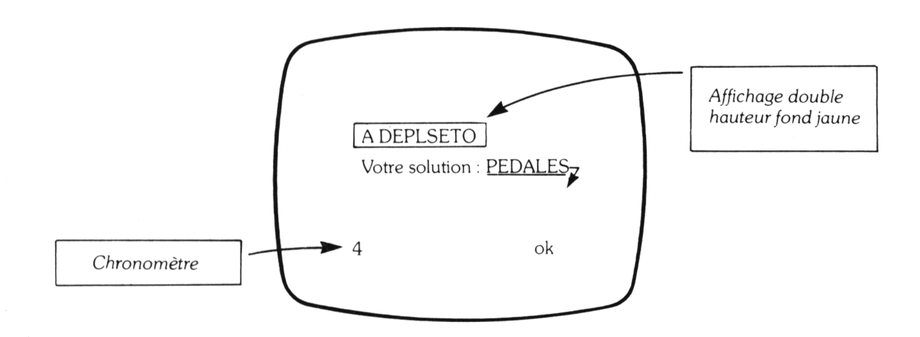
LE MOT LE PLUS LONG

Nous présentons ici une version simplifiée du jeu **'le mot le plus long'.**
Dans chaque ligne de 'DATA' sont définis :

- Le mot le plus long;
- Des lettres complémentaires (9 — longueur du mot le plus long);
- Les autres mots possibles.



Nous affichons les lettres du mot dans le désordre puis nous attendons pendant 30 secondes le résultat du joueur. Si la réponse n'est pas bonne, nous affichons la (ou les) solution(s).



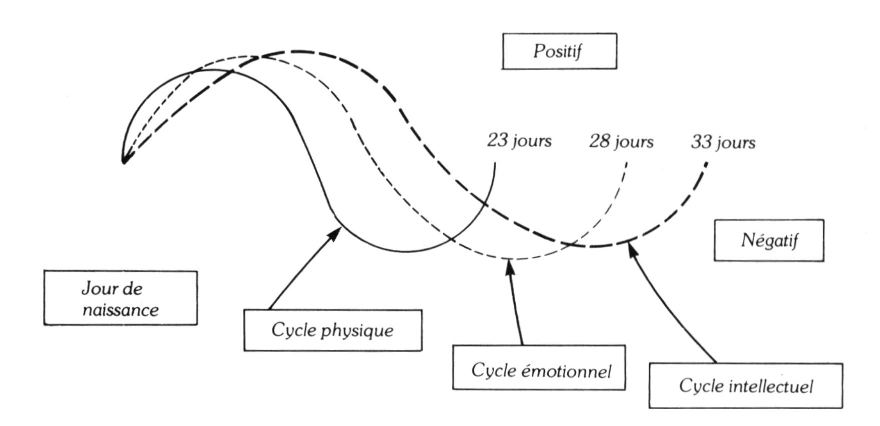
Le mot le plus long

```
10 REM  MOT    MOT LE PLUS LONG
20 :
22 INK1
25 NM=13               :REM NOMBRE DE LIGNES
30 DATA FICHIERS,W,*
40 DATA PEDALES,TO,PETALES,PEDALOS,*
50 DATA ARMOIRES,I,*
60 DATA MOQUETTE,A,MAQUETTE,*
70 DATA FAUTEUIL,Y,*
80 DATA BUREAUX,QZ,*
90 DATA CASSETTE,D,*
100 DATA RIDEAUX,HW,*
110 DATA ARDOISE,FO,FROIDES,*
120 DATA TABLEAU,BD,*
130 DATA CHAMBRE,AU,CHAMEAU,*
140 DATA CLAVIER,QZX,*
150 DATA CHEVELU,AR,CHALEUR,CHAULER,*
890 REM--------------------
900 RESTORE
910 FOR I=1 TO 9:F(I)=0:NEXT I
1000 X=RND(1)*NM
1010 FOR I=1 TO X
1015 :IF X=0 THEN 1040
1020 : READ X$
1030 : READ X$:IF X$<>"*" THEN 1030
1040 NEXT I
1050 :
1100 READ MOT$
1105 READ C$:MP$=MOT$+C$
1110 REM------------------- Affichage lettres  double hauteur fond
jaune
1115 CLS :PRINT:PRINT:PRINT:PRINT " ";CHR$(4);CHR$(27);"J";
1116 PRINT "      ";CHR$(27);"S";
1117 PRINT CHR$(27);"@";
1120 FOR I=1 TO 9
1130 : P=1+RND(1)*9
1140 : IF F(P)=0 THEN PRINT MID$(MP$,P,1);" ";:F(P)=1 ELSE 1130
1150 NEXT I
1160 PRINT CHR$(27);"W";
1165 PRINT CHR$(4)           ' Fin double hauteur
1170 :
1200 PRINT:PRINT:PRINT
1210 PLOT 5 ,10,"VOTRE SOLUTION:"
1211 S$=""
1212 REM------------------ Attente reponse -----------------
1213 T=0                      ' Chrono
1214 :
1215 X$=KEY$:IF X$<>"" THEN 1220
1216 T=T+1:IF T>500  THEN 1290 ELSE PLOT 4,23, STR$(INT(T/20 )):GOT
O 1215
1217 :
1218 :
```

```
1219 REM------ Saisie reponse
1220 S$=X$:PLOT 20+LEN(S$),10,X$
1221 :
1222 X$=KEY$:IF X$="" THEN 1222
1223 IF ASC(X$)=13 THEN 1230
1224 IF ASC(X$)<>127 THEN 1226
1225 S$=LEFT$(S$,LEN(S$)-1):PLOT 20+LEN(S$)+1,10," ":GOTO 1222
1226 S$=S$+X$:PLOT 20+LEN(S$),10,X$
1227 GOTO 1222
1228 :
1229 :
1230 IF MOT$=S$ THEN PLOT 20,20,"OK...":WAIT 300:GOTO 900
1234 :
1235 FOR I=1 TO 4
1236 :READ A$(I):IF A$(I)="*" THEN 1290
1237 :IF A$(I)=S$ THEN PLOT 20,20,"OK...":WAIT 300:GOTO 900
1238 NEXT I
1240 :
1290 X$="MA SOLUTION:"+MOT$
1300 PLOT 10,20,X$
1310 FOR I=1 TO 4
1312 : IF A$(I)="*" THEN 1400
1314 : X$=A$(I):PLOT 22,20+I,A$(I)
1315 NEXT I
1390 :
1400 WAIT 300
1410 GOTO 900
```

BIORYTHME

Il existerait chez l'homme des périodes 'positives' et des périodes 'négatives' réglées par des 'horloges internes' indépendantes de l'environnement extérieur.

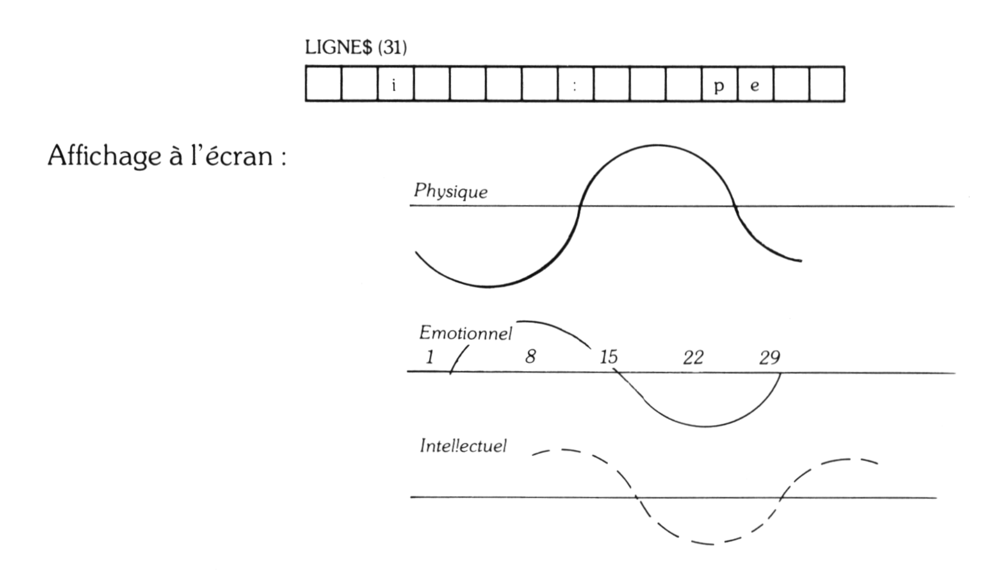


Il y aurait trois types de cycles :

- Cycle **physique** de 23 jours ;
- Cycle **émotionnel** de 28 jours ;
- Cycle **intellectuel** de 33 jours.

On pourrait ainsi connaître à l'avance les jours favorables et défavorables.

Pour afficher une ligne, nous stockons dans une table tampon LIGNE$(), les caractères à afficher.



Biorythme

```
10 REM BIOR      BIORYTHME (ecran ou imprimante)
20 :
50 DIM LIGNE$(31),MOIS$(12),JM(12)
55 :
60 DATA JAN,FEV,MARS,AVRIL,MAI,JUIN,JUILLET,AOUT,SEPT,OCT,NOV,DEC
70 FOR I=1 TO 12:READ MOIS$(I):NEXT I
90 DATA 31,28,31,30,31,30,31,31,30,31,30,31
100 FOR I=1 TO 12:READ JM(I):NEXT I
110 :
120 CP=23:CE=28:CI=33        ' Physique/Emotionnel/Intellectuel
130 :
135 CLS:PRINT "B I O R Y T H M E":PRINT
140 INPUT "VOTRE NOM ";NOM$
145 PRINT:PRINT "NAISSANCE:":PRINT
150 INPUT " JOUR,MOIS,ANNEE ";JN,MN,AN
155 IF AN<1900 THEN 150
160 J=JN:M=MN:A=AN:GOSUB 1500:XV=JV
162 GOSUB 2000
164 :
165 PRINT:PRINT "BIORYTHME:":PRINT
170 INPUT " MOIS,ANNEE ";MB,AB:IF AB<1900 THEN 170
180 J=1:M=MB:A=AB:GOSUB 1500:NJOUR=JV-XV  :REM nb de jours vecus
205 GOTO 4900          :REM Supprimer pour imprimante
210 REM============= EDITION IMPRIMANTE ===
300 K=0:CLS
325 LPRINT:LPRINT "B I O R Y T M E ":LPRINT
330 LPRINT NOM$;" ";" Ne le ";JR$(JS); " ";  JN;"/";MN;"/";AN;
340 LPRINT "  (";NJOUR;" jours )":LPRINT
350 :
400 FOR I=1 TO 31:LIGNE$(I)=" ":NEXT I
405 LPRINT:LPRINT MOIS$(MB);" ";AB:LPRINT
407 LPRINT "      negatif          positif":LPRINT
410 REM-----
420 FOR D=1 TO JM(MB)
430 :LIGNE$(16)=":"
440 :P1=INT(14*SIN((K+NJOUR)*2*PI/CP)+15.5)
450 :P2=INT(14*SIN((K+NJOUR)*2*PI/CE)+15.5)
460 :P3=INT(14*SIN((K+NJOUR)*2*PI/CI)+15.5)
500 :LIGNE$(P1)="p":LIGNE$(P2)="e":LIGNE$(P3)="i"
505 :IF P1=P2 THEN LIGNE$(P1)="*"
506 :IF P2=P3 THEN LIGNE$(P2)="*"
507 :IF P1=P3 THEN LIGNE$(P1)="*"
510 :
590 :LPRINT D;TAB(16);
600 :FOR X=1 TO 31:LPRINT LIGNE$(X);:NEXT X:LPRINT
610 :LIGNE$(P1)=" ":LIGNE$(P2)=" ":LIGNE$(P3)=" "
620 :K=K+1
630 NEXT D
710 PRINT
715 MB=MB+1:IF MB>12 THEN MB=1:AB=AB+1
720 GOTO 400
742 :
```

```
1490 REM--------------------------------------------- CALENDRIER PTT
1495 REM J,M,A --> JV
1500 MX =INT(.6+1/M+.001):MY=M+12*MX :AX=A-MX
1510 JV=J+INT((367*(MY-1)+5)/12+.001)+INT(365.25*(AX+4712)+.001)
1520 JV=JV-INT(AX/100)+INT(AX/400)
1530 RETURN
1990 REM---------------------- JOUR DE LA SEMAINE ----------
1996 REM   JV --> JS
1997 :
2000 JR$(1)="DIMANCHE":JR$(2)="LUNDI":JR$(3)="MARDI":JR$(4)="MERCRE
DI"
2010 JR$(5)="JEUDI":JR$(6)="VENDREDI":JR$(7)="SAMEDI"
2100 JS=(JV-1720977)/7
2110 JS=INT(7*(JS-INT(JS))+.001)+1
2130 RETURN
4900 REM========================= ECRAN ======
4910 HIRES:CURSET 0,0,3:FILL 60,1,6:FILL 60,1,7:FILL 60,1,1
4920 CURSET 0,30,1:DRAW 230,0,1
4930 CURSET 0,90,1:DRAW 230,0,1
4940 CURSET0,150,1:DRAW 230,0,1
4950 DT$="  1      8     15      22      29"
4960 CURSET 1,81,3
4970 FOR I=1 TO LEN(DT$):X=ASC(MID$(DT$,I,1)):CHAR X,0,1:CURMOV 7,0
,3
4983 NEXT I
4984 MS$="Physique":XM=20:YM=10:GOSUB 4991
4985 MS$="Emotionnel":XM=20:YM=70 :GOSUB 4991
4986 MS$="Intellect ":XM=20:YM=130:GOSUB 4991
4987 GOTO 5000
4990 :
4991 CURSET XM,YM,3
4992 FOR I=1 TO LEN(MS$)
4993 :X=ASC(MID$(MS$,I,1)):CHAR X,0,1:CURMOV 6,0,3
4994 NEXT I
4995 RETURN
5000 FOR D=1 TO JM(MB)
5040 :P1=25*SIN((K+NJOUR)*2*PI/CP)
5050 :P2=25*SIN((K+NJOUR)*2*PI/CE)
5060 :P3=25*SIN((K+NJOUR)*2*PI/CI)
5100 :X=D*7+7
5110 :CURSET X,30-P1,1:DRAW 4,0,1
5120 :CURSET X,90-P2,1:DRAW 4,0,1
5130 :CURSET X,150-P3,1:DRAW 4,0,1
5200 :K=K+1
5210 NEXT D
```

```
B I O R Y T M E
GIBET  Ne le JEUDI 13 /12 /1945   (13745  Jours )
AOUT 1983
     negatif         positif

1       *       i :
2     p    e i    :
3   p      i  e   :
4  p    i        e:
5  p  i           : e
6    *            :    e
7   i  p          :       e
8  i      p       :         e
9  i         p    :           e
10 i             p:            e
11  i             :  p         e
12    i           :     p      e
13     i          :        p  e
14       i        :         ep
15          i     :       e    p
16            i   :    e       p
17               i: e         p
18               e: i       p
19            e   :   i   p
20         e      :    p i
21      e         :p       i
22    e        p  :         i
23  e      p      :           i
24 e    p         :            i
25 e  p           :            i
26 ep             :            i
27 pe             :           i
28 p  e           :          i
29   p  e         :         i
30     p   e      :       i
31        p   e   :    i

SEPT 1983

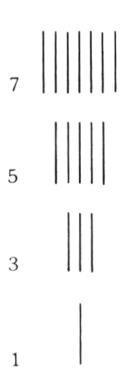
     negatif         positif

1            p   e:  i
2                pi e
3               i :  p e
4            i    :     p e
5          i      :        pe
6       i         :          pe
7     i           :            *
8    i            :            *
9   i             :           pe
10 i              :         p e
11 i              :       p e
12 i              :    p  e
13  i             :p   e
14    i        p  : e
15     i   p     e:
```

JEU DE MARIENBAD

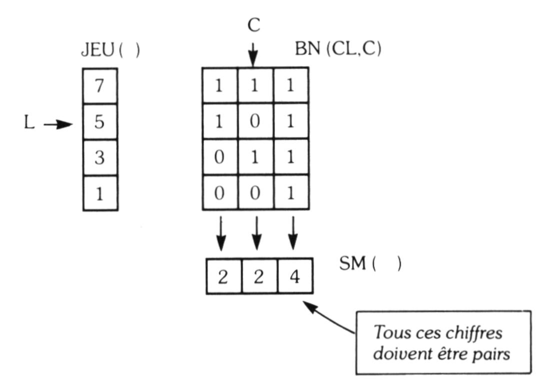
Règle du jeu

Quatre rangées contiennent 7, 5, 3, 1 allumettes. Chaque joueur doit prendre à tour de rôle des allumettes dans une seule rangée à la fois. Celui qui prend la dernière perd.



Un algorithme donne le nombre d'allumettes à prendre pour laisser un jeu perdant à l'adversaire.

- On convertit en binaire le nombre d'allumettes de chaque ligne (dans une table BN(,)).
- On effectue la somme de chaque colonne de la table BN(,).



- Pour laisser un jeu perdant à l'adversaire, il suffit que les chiffres de toutes les colonnes de SM() soient pairs.

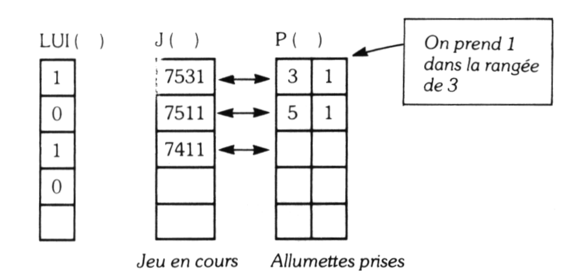
Jeu de Marienbad

```
10 REM   MAR      JEU DE MARIENBAD
20 :
25 CLS
30 REM--------------- Jeu depart
100 JEU(1)=7:JEU(2)=5
110 JEU(3)=3:JEU(4)=1
140 :
150 GOSUB 4000
160 REM-------------
165 PRINT
170 INPUT "Voulez vous commencer (O/N) ";R$
175 JR=0:IF R$="O" THEN JR=1
176 :
177 FP=0
178 :
180 IF JR=1 THEN GOSUB 200:JR=1-JR
185 IF FP=1 THEN 100
190 IF JR=0 THEN GOSUB 300:JR=1-JR
192 IF FP=1 THEN 100
195 GOTO 180
200 REM================================= JOUEUR ===============
210 PRINT:INPUT "QUELLE LIGNE ";L
220 INPUT "COMBIEN       ";N
230 :
235 IF N>JEU(L) THEN PRINT:PRINT "VOUS TRICHEZ":PRINT:GOTO 210
240 JEU(L)=JEU(L)-N
250 :
260 GOSUB 4000                 ' Affichage
265 GOSUB 5000                 ' Fin de partie
267 IF S=1 THEN PRINT "VOUS AVEZ GAGNE":FP=1
270 RETURN
290 REM================================== MACHINE =============
300 FOR L=1 TO 4
310 : X=JEU(L):GOSUB 1000
320 NEXT L
400 :
490 REM------------ Recherche jeu gagant -----------------------
500 FOR L=1 TO 4               ' 4 lignes
510 : FOR N=1 TO JEU(L)        ' On essaie d'en prendre 1 a JEU(L)
515 :  IF JEU(L)=0 THEN 570
520 :  X=JEU(L)-N:GOSUB 1000
530 :  GOSUB 2000              ' Sommation colonnes
540 :  IF GN=1 THEN GOTO 680
550 : NEXT N
560 : X=JEU(L):GOSUB 1000      ' Retablissement de la ligne binaire
570 NEXT L
```

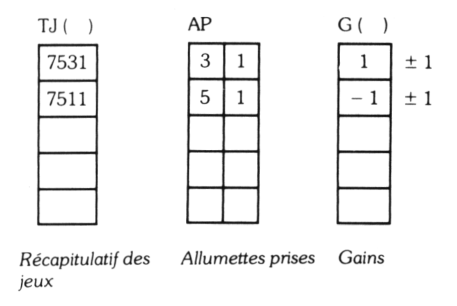
```
580 REM---------- Pas de solution gagnante ------
590 FOR L=1 TO 4
600 : IF JEU(L)>0 THEN N=1:GOTO 690
610 NEXT L
620 STOP
670 :
680 IF SM(0)=2 AND SM(1)=0 AND SM(2)=0 AND N=JEU(L) THEN N=JEU(L)-1
:GOTO 700
690 IF SM(0)=2 AND SM(1)=0 AND SM(2)=0 THEN N=JEU(L)
692 IF SM(0)=0 AND SM(1)=0 AND SM(2)=0 THEN N=JEU(L)-1
695 :
700 PRINT:PRINT "JE JOUE: LIGNE:";L;"J'EN PRENDS:";N
710 JEU(L)=JEU(L)-N
715 :
720 GOSUB 4000
730 :
740 GOSUB 5000
745 IF S=1 THEN PRINT:PRINT "J'AI GAGNE":PRINT:FP=1
750 RETURN
990 REM------------ CV BINAIRE
1000 REM   ENTREE: X,L
1010 REM   SORTIE: REMPLIT BN(L,)
1020 :
1030 FOR P=2 TO 0 STEP-1
1040 : BN(L,P)=INT(X/(2^P)+.01)
1050 : X=X-BN(L,P)*(2^P)
1060 NEXT P
1070 RETURN
1990 REM------------ SOMMATION ------------
2000 FOR C=2 TO 0 STEP-1
2005 : S=0
2010 : FOR L1=1 TO 4
2020 :  S=S+BN(L1,C)
2030 : NEXT L1
2040 : SM(C)=S
2050 NEXT C
2060 :
2090 GN=0
2100 FOR C=2 TO 0 STEP-1
2110 : IF INT(SM(C)/2)<>SM(C)/2 THEN GN=0:RETURN
2120 NEXT C
2130 GN=1
2140 RETURN
3990 REM ------------------- DESSIN
4000 PRINT
4010 FOR L=1 TO 4
4012 :PRINT L;" ";
4020 :FOR N=1 TO JEU(L)
4025 : IF JEU(L)=0 THEN 4040
4030 : PRINT "I";
4040 :NEXT N
4050 PRINT
4060 NEXT L
4070 RETURN
4990 REM-------------- FIN DE PARTIE?
5000 S=0:FOR L=1 TO 4:S=S+JEU(L):NEXT L:RETURN
```

JEU DE MARIENBAD PAR APPRENTISSAGE

Nous proposons le jeu de **Marienbad.** Mais cette fois-ci, la machine apprend à jouer en 'imitant' son adversaire.
Pour la première partie, la machine prend une allumette dans une rangée quelconque.
Dans 2 tables J() et P() sont stockés tous les jeux de la partie.



A la fin de la partie nous cumulons les jeux de la partie achevée dans 3 tables.



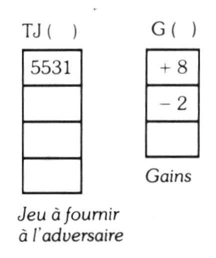
Dans une table des gains G(), nous faisons +1 si la partie a été gagnée et −1 si elle a été perdue. Ainsi, pour les jeux suivants, la machine consulte ces tables pour chercher des configurations gagnantes.

Au fur et à mesure des parties, la machine dispose de plus en plus de configurations gagnantes. Elle doit jouer de mieux en mieux, surtout si son adversaire joue bien.
Pour tester le programme, laissez-le commencer la partie la première fois et gagnez la partie. Commencez la seconde partie (comme lui) et il jouera comme vous aviez joué lors de la première partie et gagnera lui aussi.

Autre méthode :

La table des jeux peut être organisée différemment :

- On stocke le jeu à fournir à l'adversaire (et non le jeu actuel).
- On recherche alors, **par essais successifs,** le nombre d'allumettes à enlever pour parvenir au jeu cherché.
- Cette méthode est plus puissante que la précédente mais plus compliquée à programmer.



Supprimer : 237 GOSUB 6000 : GOSUB 6140
Ajouter : 245 GOSUB 6000 : GOSUB 6140

Les instructions 290 à 589 deviennent :

```
290 REM================================== MACHINE =======
300 :
400 MCOUP=0
410 REM-------------------- Recherche table des jeux
415 IF TJ(1)=0 THEN N=1:GOTO 600
420 FOR L1=1 TO 4
425 :FOR N1=1 TO JEU(L1)
427 : SJ=JEU(L1)
430 : JEU(L1)=JEU(L1)-N1:GOSUB 6000
440 : FOR I=1 TO 200
445 :   IF TJ(I)=0 THEN 470
450 :   IF JCOD<>TJ(I) THEN 460
455 :   IF G(I)>MCOUP THEN MCOUP=G(I):NJ=N1:PL=L1
460 : NEXT I
470 : JEU(L1)=SJ
480 :NEXT N1
490 NEXT L1
510 :
520 IF MCOUP=0 THEN N=1:GOTO 600
545 L=PL:N=NJ
560 :
580 PRINT:PRINT "J'EN PRENDS:";N;" LIGNE:";L
582 JEU(L)=JEU(L)-N
583 GOSUB 6000:GOSUB 6140
584 :
585 GOSUB 4000
586 GOSUB 5000
587 IF S=1 THEN PRINT :PRINT "J'AI GAGNE":PRINT:G=1:FP=1
589 RETURN
```

Jeu par apprentissage — Marienbad

```
10 REM  APPR  JEU PAR APPRENTISSAGE -MARIENBAD -
20 :
22 DIM TJ(200),AP(200),G(200),J(30),P(30),LUI(30)
25 CLS
30 :REM-------------- Jeu depart
100 JEU(1)=7:JEU(2)=5
110 JEU(3)=3:JEU(4)=1
120 :
150 GOSUB 4000
160 REM--------------
165 PRINT
170 INPUT "Voulez vous commencer (O/N) ";R$
175 JR=0:IF R$="O" THEN JR=1
176 :
177 FP=0:COUP=0
178 :
180 IF JR=1 THEN GOSUB 200:JR=1-JR
185 IF FP=1 THEN GOSUB 7000:GOTO 100
190 IF JR=0 THEN GOSUB 300:JR=1-JR
192 IF FP=1 THEN GOSUB 7000:GOTO 100
195 GOTO 180
200 REM================================== JOUEUR ========
210 PRINT:INPUT "QUELLE LIGNE ";L
220 INPUT "COMBIEN      ";N
230 :
235 IF N>JEU(L) THEN PRINT:PRINT "VOUS TRICHEZ":PRINT:GOTO 210
237 GOSUB 6000:GOSUB 6140
239 :
240 JEU(L)=JEU(L)-N
250 :
260 GOSUB 4000
265 GOSUB 5000
267 IF S=1 THEN PRINT "VOUS AVEZ GAGNE":FP=1:G=-1
270 RETURN
290 REM=================================== MACHINE =======
300 :
310 GOSUB 6000          :REM conversion -> jeu code
320 JAC=JCOD            :REM EX:JCOD=7531
330 :
400 MCOUP=0
410 REM-------------------- Recherche table des jeux
420 FOR I=1 TO 200
425 :IF TJ(I)=0 THEN 520
430 :IF JAC<>TJ(I) THEN 500
440 :IF G(I)>MCOUP THEN MCOUP=G(I):PCOUP=I
500 NEXT I
510 :
520 IF MCOUP=0 THEN N=1:GOTO 600
```

```
530 REM------------- decodage: 31 --> 3 et 1
540 NL=INT(AP(PCOUP)/10):N=AP(PCOUP)-NL*10
542 FOR I=1 TO 4
543 :IF NL=JEU(I) THEN L=I:GOTO 560
544 NEXT I
545 STOP
550 :
555 :
556 :
557 :
560 GOSUB 6000:GOSUB 6140
570 :
580 PRINT:PRINT "J'EN PRENDS:";N;" LIGNE:";L
582 JEU(L)=JEU(L)-N
583 :
584 :
585 GOSUB 4000
586 GOSUB 5000
587 IF S=1 THEN PRINT :PRINT "J'AI GAGNE":PRINT:G=1:FP=1
589 RETURN
590 REM-------- PAS DE SOLUTION EN TABLE
600 N1=0
610 FOR I=1 TO 4
620 :IF JEU(I)>N1 THEN N1=JEU(I):L=I
630 NEXT I
640 :
650 GOTO 560
1075
3990 REM -------------------- DESSIN
4000 PRINT
4010 FOR L=1 TO 4
4012 :PRINT L;" ";
4020 :FOR N=1 TO JEU(L)
4025 : IF JEU(L)=0 THEN 4040
4030 : PRINT "I";
4040 :NEXT N
4050 PRINT
4060 NEXT L
4070 RETURN
4990 REM------------- FIN DE PARTIE?
5000 S=0
5005 FOR L=1 TO 4
5010 :S=S+JEU(L)
5020 NEXT L
5030 RETURN
```

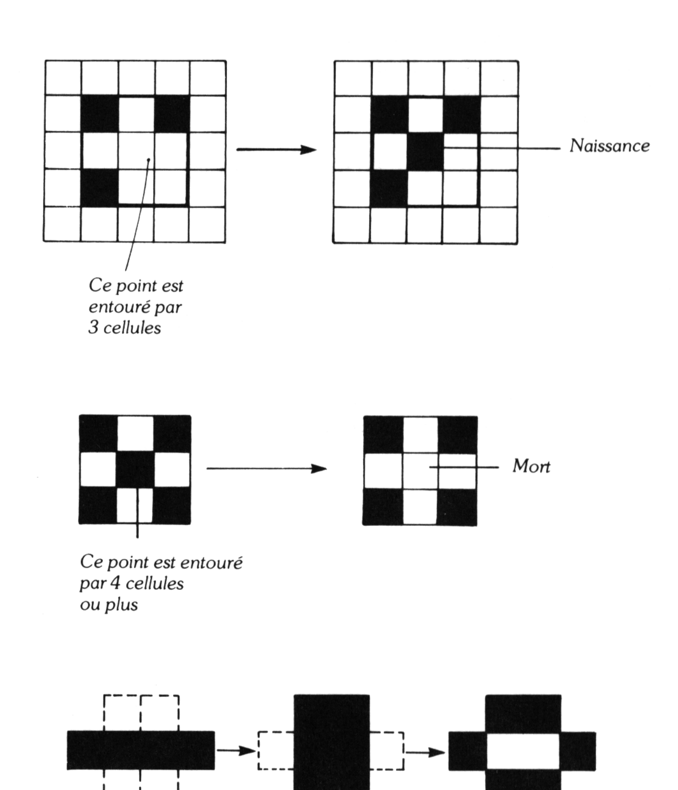
```
5990 REM---------- CONVERSION JEU( )--> JEU CODE --
6000 :
6010 FOR I=1 TO 4:X(I)=JEU(I):NEXT I
6020 :
6030 IV=0
6040 FOR I=1 TO 3
6050 :IF X(I+1)<X(I) THEN X=X(I):X(I)=X(I+1):X(I+1)=X:IV=1
6060 NEXT I
6070 IF IV=1 THEN 6030
6080 :
6100 JCOD=0
6110 FOR I=1 TO 4
6120 :JCOD=JCOD+X(I)*10^(I-1)
6130 NEXT I
6135 RETURN
6140 REM------- MAJ TABLE JEU ACTUEL
6150 COUP=COUP+1:J(COUP)=JCOD
6160 P(COUP)=JEU(L)*10+N
6170 LUI(COUP)=JR
6180 RETURN
6990 REM--------------- MAJ FIN DE PARTIE
7000 :
7010 FOR I=1 TO COUP
7020 : X=J(I):Y=P(I)
7030 : FOR J=1 TO 1000
7035 :  IF TJ(J)=0 THEN TJ(J)=X:AP(J)=Y
7040 :  IF X=TJ(J) THEN IF AP(J)=Y THEN 7080
7050 : NEXT J
7060 :
7080 : IF LUI(I)=1 THEN G(J)=G(J)-G
7090 : IF LUI(I)=0 THEN G(J)=G(J)+G
7110 :
7200 NEXT I
7210 RETURN
```

LE JEU DE LA VIE

CONWAY a imaginé le **'jeu de la vie'.** Des cellules se reproduisent ou meurent en fonction de leur juxtaposition.

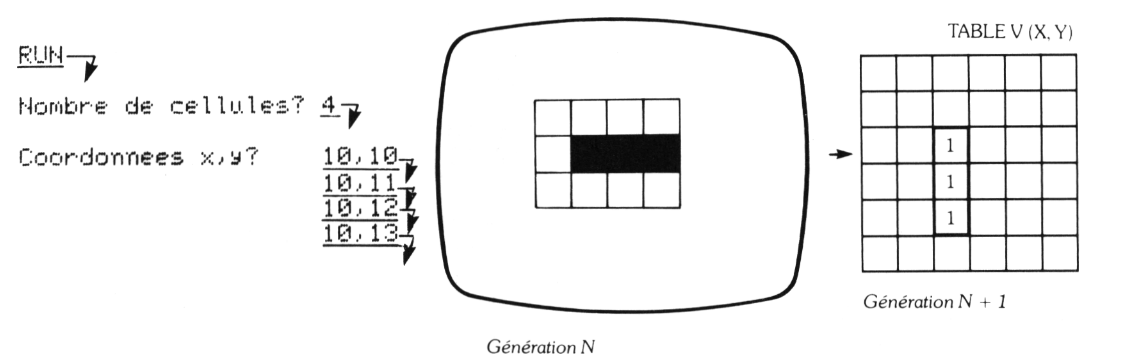
Les règles sont les suivantes :

- Chaque point entouré par 3 cellules donne lieu à une naissance.
- Chaque cellule entourée de 4 cellules ou plus meure (étouffement).
- Chaque cellule entourée par moins de 2 cellules meure (isolement).



Première méthode :

Nous appliquons directement le principe énoncé : Pour chaque point, nous considérons les points voisins. Nous positionnons à 1 les éléments d'une table V(X,Y) lorsqu'il y a naissance puis nous affichons le contenu de la table.

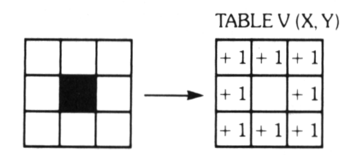


Génération N

Deuxième méthode (d'après revue Byte) :

La première méthode présentée est très lente puisque pour chaque point, nous explorons systématiquement ses 8 points voisins. Sachant que seuls les points allumés peuvent provoquer une naissance ou une mort dans leur voisinage, nous procédons ainsi :

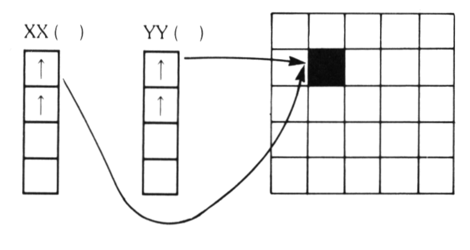
- Pour chaque point allumé, nous augmentons de 1 les 8 points voisins.



- Nous explorons la table V(X,Y) et nous allumons tous les points totalisant 3 (naissance). Nous éteignons tous les points supérieurs à 4.

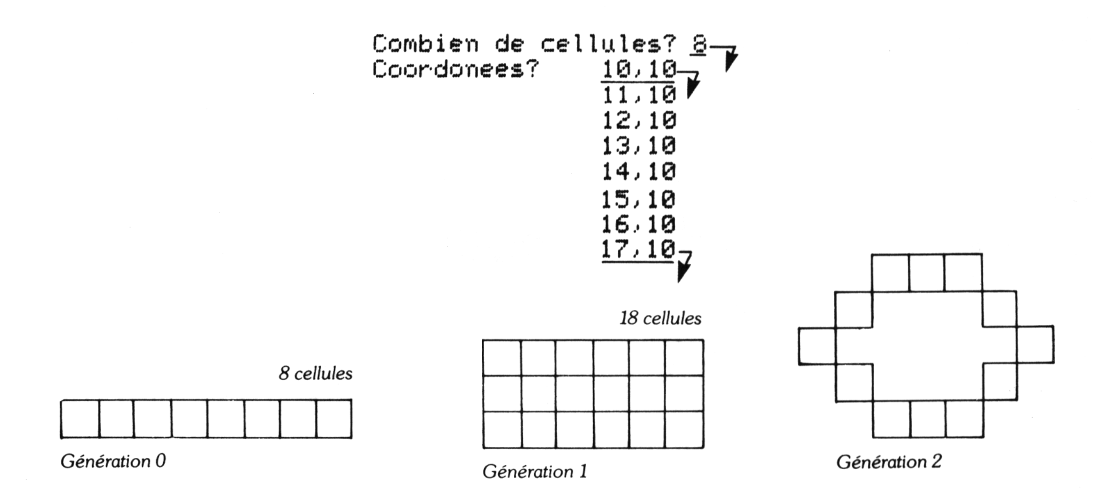
Troisième méthode :

De façon à ne s'intéresser qu'aux points allumés, nous ajoutons 2 tables XX() et YY() "pointant" vers les cellules en vie. Ainsi le temps de traitement est notablement réduit. Il est indépendant de la dimension de la table V(X,Y)



Pointeurs

VOICI UN EXEMPLE D'EXECUTION :



A la 48e génération, nous obtenons une combinaison stable.
En travaillant en haute résolution, le territoire devient beaucoup plus important.
Jl suffit de remplacer : LORES par HIRES
PLOT X,Y,16 par CURSET X,Y,0
PLOT X,Y,10 par CURSET X,Y,1
SCRN (X,Y) par POINT (X,Y)=−1

```
RUN
Nombre de cellules? 15
Coordonnees x,y?     40,30
                     40,31
                     40,32
                       :
                     40,44
```

Jeu de la vie n° 1

```
10 REM VIE1   JEU DE LA VIE NO1
20 :
30 CLS
40 X1=2:X2=20                    :REM limites territoire
50 Y1=2:Y2=20
60 DIM V(X2,Y2)
70 :
80 INPUT "Combien de cellules  ";NC
85 PRINT
90 FOR N=1 TO NC
100 :INPUT "COORDONNEES X,Y ";X,Y
110 :V(X,Y)=1
120 NEXT N
130 GOSUB 500
140 REM------------------------------------
150 FOR Y=Y1 TO Y2
160 :FOR X=X1 TO X2
170 :
180 : V(X,Y)=0:TV=0
190 :
200 : FOR DY=-1 TO 1          :REM 8 points voisins
210 :  FOR DX=-1 TO 1
220 :    IF DX=0 THEN IF DY=0 THEN 240
230 :    IF SCRN(X+DX,Y+DY)=18 THEN TV=TV+1
240 :   NEXT DX
250 :  NEXT DY
260 :  IF TV=>4 THEN V(X,Y)=0:GOTO 300
270 :  IF TV=3 THEN V(X,Y)=1:GOTO 300
280 :  IF TV=2 OR TV=3 THEN IF SCRN(X,Y)=18 THEN V(X,Y)=1
290 :
300 :NEXT X
310 NEXT Y
320 :
330 GN=GN+1                   :REM No generation
340 GOSUB 500
345 GOTO 150
350 :
500 REM------------ AFFICHAGE ---------------
510 LORES0
520 :
530 FOR Y=Y1 TO Y2
540 :FOR X=X1 TO X2
550 : IF V(X,Y)=1 THEN PLOT X,Y,18
560 :NEXT X
570 NEXT Y
580 :
590 X$="GENERATION:"+STR$(GN)
600 PLOT 25,20,X$
610 RETURN
```

Jeu de la vie n° 2

```
10 REM VIE2      JEU DE LA VIE   NO2
20 :
30 CLS
40 X1=2:X2=23                  :REM limites territoire
50 Y1=2:Y2=22
60 DIM V(X2,Y2)
70 :
80 INPUT "Combien de cellules ";NC
85 PRINT
90 FOR N=1 TO NC
100 :INPUT "Coordonnees X,Y ";X,Y
110 :V(X,Y)=3
120 NEXT N
124 LORES0
125 GOSUB 500
129 REM------
140 :
150 FOR Y=Y1 TO Y2
160 :FOR X=X1 TO X2
170 :
180 :  IF SCRN(X,Y)<>18 THEN 300
190 :
200 :  FOR DY=-1 TO 1         :REM 8 Points voisins
210 :   FOR DX=-1 TO 1
220 :     IF DX=0 THEN IF DY=0 THEN 240
230 :      V(X+DX,Y+DY)=V(X+DX,Y+DY)+1
240 :    NEXT DX
250 :   NEXT DY
290 :
300 :NEXT X
310 NEXT Y
320 :
330 GN=GN+1
340 GOSUB 500
345 GOTO 150
350 :
500 REM------------ AFFICHAGE ---------------
510 :
520 :
530 FOR Y=Y1 TO Y2
540 :FOR X=X1 TO X2
545 :  IF V(X,Y)<2 THEN PLOT X,Y,16:GOTO 560
550 :IF V(X,Y)=3 THEN PLOT X,Y,18:GOTO560
555 :  IF V(X,Y)=>4 THEN PLOT X,Y,16
559 :
560 :  V(X,Y)=0
565 NEXT X
570 NEXT Y
580 :
590 X$="GENERATION:"+STR$(GN)
600 PLOT 25,20,X$
610 RETURN
```

Jeu de la vie n° 3

```
10 REM    JEU DE LA VIE NO 3
20 :
25 DIM XX(99),YY(99)
26 DIM X1(99),Y1(99)
30 CLS
40 X1=2:X2=30
50 Y1=2:Y2=25
60 DIM V(X2+1,Y2+1)
70 :
80 INPUT "COMBIEN DE CELLULES ";NC
85 PRINT
90 FOR N=1 TO NC
100 :INPUT "COORDONNEES X,Y ";X,Y
110 :V(X,Y)=3
112 :XX(N)=X:YY(N)=Y
120 NEXT N
124 LORES0
125 GOSUB 500
129 REM------
140 :
150 FOR N=1 TO NC
160 :X=XX(N):Y=YY(N)
170 :
200 : FOR DY=-1 TO 1
205 :  IF Y+DY<Y1 OR Y+DY>Y2 THEN 250
210 :  FOR DX=-1 TO 1
215 :   IF X+DX<X1 OR X+DX>X2 THEN 240
220 :
230 :   V(X+DX,Y+DY)=V(X+DX,Y+DY)+1
240 :   NEXT DX
250 :  NEXT DY
290 :
300 NEXT N
310 :
330 GN=GN+1
340 GOSUB 500
345 GOTO 150
```

```
500 REM------------ AFFICHAGE ---------------
510 :
515 N1=0
520 :
530 FOR N=1 TO NC
540 :XX=XX(N):YY=YY(N)
541 :FOR DY=-1 TO 1:FOR DX=-1 TO 1
542 : X=XX+DX:Y=YY+DY
543 : IF V(X,Y)=0 THEN 565
545 : IF V(X,Y)<=2 THEN PLOT X,Y,16:GOTO 565
550 :IF V(X,Y)=3 THEN PLOT X,Y,18:GOTO 560
555 : IF V(X,Y)=>5 THEN PLOT X,Y,16:GOTO 565
558 :IF SCRN(X,Y)=18 THEN IF V(X,Y)=4 THEN 560
559 : GOTO 565
560 :
562 :N1=N1+1:X1(N1)=X:Y1(N1)=Y
564 :
565 : V(X,Y)=0
567 :NEXT DX:NEXT DY
570 NEXT N
580 :
585 NC=N1
587 FOR N=1 TO NC:YY(N)=Y1(N):XX(N)=X1(N):NEXT N
590 X$="GENERATION:"+STR$(GN)+STR$(NC)
600 PLOT 25,20,X$
610 RETURN
```

Jeu de la vie haute résolution

```
10 REM VIE4     JEU DE LA VIE HAUTE RESOLUTION
20 :
25 DIM XX(120),YY(120)
26 DIM X1(120),Y1(120)
30 CLS
40 X1=2 :X2=70
50 Y1=2 :Y2=70
60 DIM V(X2+1,Y2+1)
70 :
80 INPUT "COMBIEN DE CELLULES ";NC
85 PRINT
90 FOR N=1 TO NC
100 :INPUT "COORDONNEES X,Y ";X,Y
110 :V(X,Y)=3
112 :XX(N)=X:YY(N)=Y
120 NEXT N
124 HIRES:INK6
125 GOSUB 500
129 REM------
140 :
150 FOR N=1 TO NC
160 :X=XX(N):Y=YY(N)
170 :
200 : FOR DY=-1 TO 1
205 :  IF Y+DY<Y1 OR Y+DY>Y2 THEN 250
210 :  FOR DX=-1 TO 1
215 :   IF X+DX<X1 OR X+DX>X2 THEN 240
220 :
230 :   V(X+DX,Y+DY)=V(X+DX,Y+DY)+1
240 :   NEXT DX
250 :  NEXT DY
290 :
300 NEXT N
310 :
330 GN=GN+1
340 GOSUB 500
342 :REM--- GOSUB 630   Pour edition imprimante
345 GOTO 150
350 :
```

```
500 REM------------ AFFICHAGE ----------------
510 :
515 N1=0
530 FOR N=1 TO NC
540 :XX=XX(N):YY=YY(N)
541 :FOR DY=-1 TO 1:FOR DX=-1 TO 1
542 : X=XX+DX:Y=YY+DY
543 : IF V(X,Y)=0 THEN 565
545 :IF V(X,Y)<=2 THEN CURSET X,Y,0:GOTO 565
550 :IF V(X,Y)=3 THEN CURSET X,Y,1:GOTO 560
555 :IF V(X,Y)=>5 THEN CURSET X,Y,0:GOTO 565
558 :IF POINT(X,Y)=-1 THEN IF V(X,Y)=4 THEN 560
559 : GOTO 565
560 :
562 :N1=N1+1:X1(N1)=X:Y1(N1)=Y
564 :
565 : V(X,Y)=0
567 :NEXT DX:NEXT DY
570 NEXT N
580 :
585 NC=N1
587 FOR N=1 TO NC:YY(N)=Y1(N):XX(N)=X1(N):NEXT N
590 PRINT "GENERATION:";GN;" CELLULES:";NC
600 :
610 RETURN
620 REM------------------- Edition imprimante
630 FOR L=10 TO 40
640 :FOR C=15 TO 60
650 : IF POINT(C,L)=-1 THEN LPRINT "*";:GOTO 670
660 : LPRINT " ";
670 : NEXT C
680 :LPRINT
690 NEXT L
695 LPRINT:LPRINT "GENE:";GN:LPRINT
700 RETURN
```

```
         *
        ***
       *****
      **   **
       *****
  **    ***    **
 *  *    *    *  *
******       ******
 *  *    *    *  *
  **    ***    **
       *****
      **   **
       *****
        ***
         *
```

Génération 7

```
          *
         ***
        * * *
        *   *
      *  * *  *
      * ***** *
       *** ***
          *

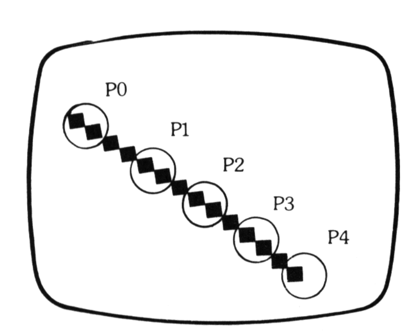
 *  **         **  *
*   *           *   *
 *  **         **  *

          *
       *** ***
      * ***** *
      *  * *  *
        *   *
        * * *
         ***
          *
```

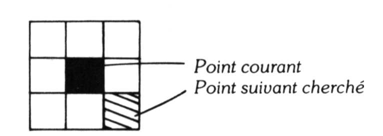
Génération 13

RECHERCHE DE LA PENTE D'UNE DROITE

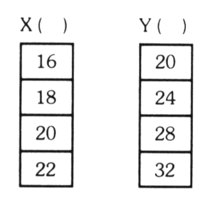
Si les jeux ne vous amusent plus, si la gestion vous ennuie, voici un domaine pour l'instant peu exploré : **la reconnaissance de formes** (sur un écran par exemple).
Soit une droite dessinée sur un écran :



- Nous voulons déterminer l'équation de cette droite.
- Nous supposons que la droite ne présente pas de discontinuité.
- Après avoir trouvé le premier point de la droite, nous recherchons tous les points de celle-ci en recherchant à partir de chaque point le point suivant.



- Les coordonnées des points sont stockées dans 2 tables X() et Y()



- Par une méthode des moindres carrés, nous calculons la pente de la droite. Un coefficient de qualité égal à $\frac{\mathbf{COVxy^2}}{\mathbf{VarX*VarY}}$ nous permet de déterminer si la courbe testée est bien une droite.

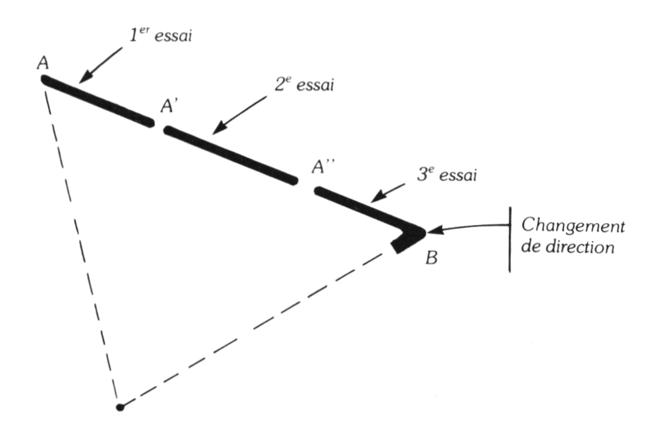
Recherche de la pente d'une droite

```
10 REM RFORM    RECHERCHE DE LA PENTE D'UNE DROITE
20 :
25 REM---------- Trace d'une droite Pour essai
30 HIRES
40 CURSET 20,3,1
50 DRAW 50,100,2
70 REM------- Recherche Premier Point
80 FOR Y=0 TO 199
90 :FOR X=0 TO 239
100 : IF POINT(X,Y)=-1 THEN 150
110 :NEXT X
120 NEXT Y
130 PRINT "ECRAN VIERGE":STOP
140 :
150 PRINT "PREMIER POINT:";X,Y
160 REM================= Recherche de Points
170 X(1)=X:Y(1)=Y            :REM Premier Point
180 XC=X:YC=Y                :REM Point courant
190 :
200 FOR PT=2 TO 5            :REM 5 Points saisis
210 :FOR DEPL=1 TO 4         :REM DePlacement entre 2 Points
220 : FOR Y1=YC-1 TO YC+1    :REM Recherche Point suivant
230 :  FOR X1=XC-1 TO XC+1
240 :   IF X1=XC THEN IF YC=Y1 THEN 270
250 :   IF Y1=YA THEN IF X1=XA THEN 270
260 :   IF POINT(X1,Y1)=-1 THEN XA=XC:YA=YC:XC=X1:YC=Y1:GOTO 310
270 :   NEXT X1
280 : NEXT Y1
290 PRINT "DISCONTINUITE":STOP
300 :
310 :NEXT DEPL
320 X(PT)=XC:Y(PT)=YC
330 NEXT PT
340 REM---
360 NP=5
370 REM------------Calcul de xmoy et ymoy
380 SX=0:SY=0:X2=0
390 FOR I=1 TO NP
400 :SX=SX+X(I):SY=SY+Y(I):X2=X2+X(I)*X(I)
410 NEXT I
420 XM=SX/NP:YM=SY/NP
```

```
430 REM------------ Calcul de cov XY var XY varX varY
440 SC=0:SV=0
450 FOR I=1 TO NP
460 :SC=SC+X(I)*Y(I)-XM*YM
470 :SV=SV+X(I)*X(I)-XM*XM
480 :S2=S2+Y(I)*Y(I)-YM*YM
490 NEXT I
500 CV=SC/NP:VX=SV/NP:VY=S2/NP
510 :
520 A=CV/VX                :REM Pente
530 B=YM-A*XM
540 RO=CV*CV/(VX*VY)
550 PRINT "PENTE:";A;" COEFF QUALITE=";RO;
560 :
570 GET X$
580 TEXT
```

Reconnaissance de figures composées de droites :

- **Comment déterminer un changement de direction?**
- Un premier essai fournit la pente de la première droite.
- Un 2e essai donne le même résultat.
- Au 3e essai, nous constatons grâce au coefficient de qualité que les points ne sont plus alignés.

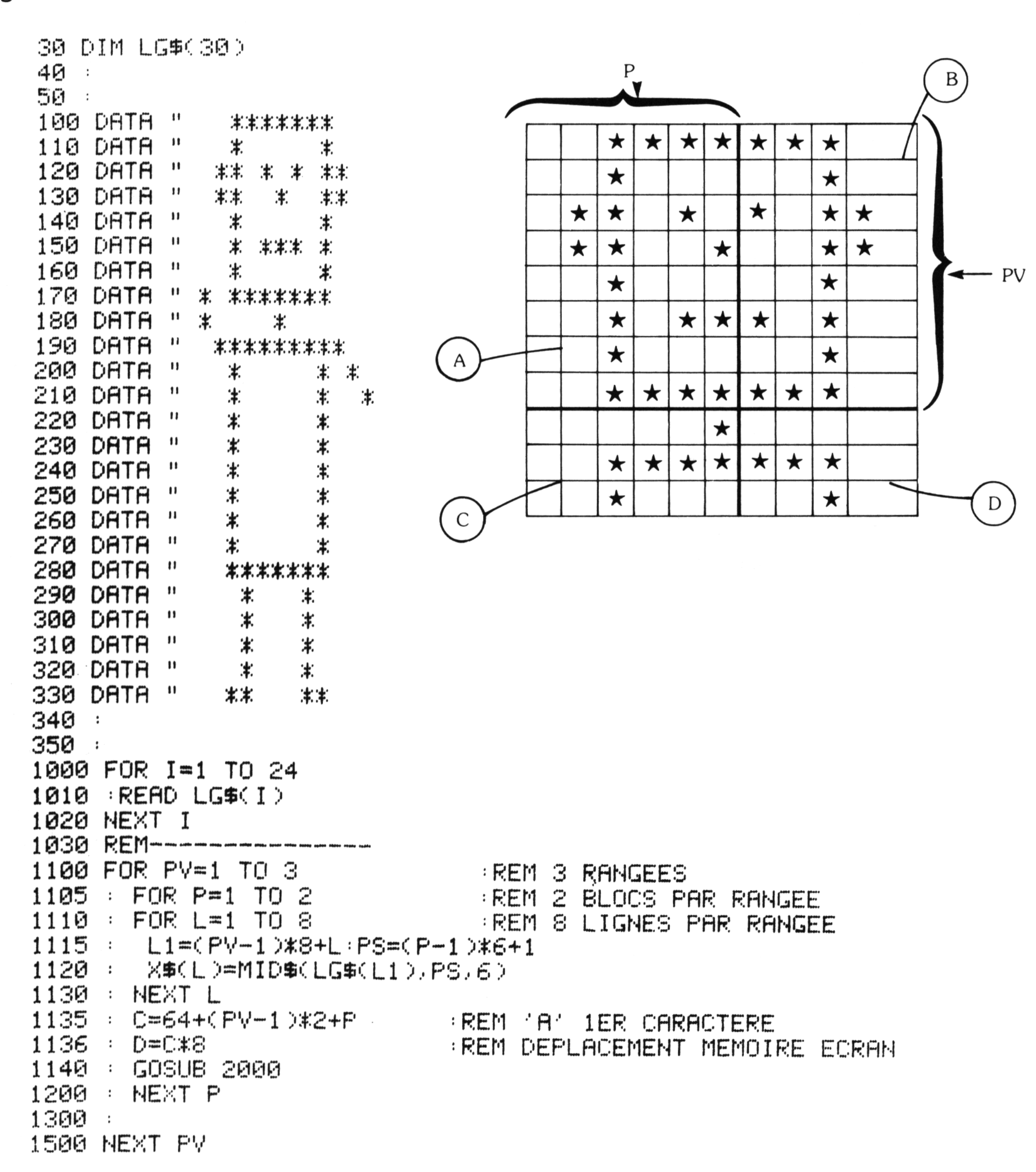


```
10 REM ROTF    ROTATION D'UNE FIGURE
20 :
22 DIM X(150),Y(150)
25 REM---------- Trace d'une figure Pour essai
30 HIRES
35 X0=20:Y0=18:L=30:H=30
37 CURSET X0,Y0,1
40 CURSET X0+L/2,Y0+H/2,1
50 DRAW -L ,0,1
55 DRAW 0  ,-H,1
56 DRAW L  ,H/2,1
57 DRAW 0,H/2,1
70 REM------- Recherche Premier Point
75 :
80 FOR Y=0 TO 199
90 :FOR X=0 TO 239
100 : IF POINT(X,Y)=-1 THEN 150
110 :NEXT X
120 NEXT Y
130 PRINT "ECRAN VIERGE":STOP
140 :
150 PRINT "PREMIER POINT:";X,Y
160 REM================= Recherche de Points
170 :
180 XC=X:YC=Y                :REM Point courant
185 XD=X:YD=Y
190 :
200 FOR P=3 TO 1000
220 : FOR Y1=YC-1 TO YC+1    :REM Recherche Point suivant
230 :  FOR X1=XC-1 TO XC+1
240 :   IF X1=XC THEN IF YC=Y1 THEN 270
250 :
255 :   IF POINT(X1,Y1)<>-1 THEN 270
256 :
260 :    FOR I=1 TO 3        :REM Retour arriere
262 :      IF X1=X(P-I) THEN IF  Y1=Y(P-I) THEN 270
265 :    NEXT I
267 :    XC=X1:YC=Y1:GOTO 310
270 :   NEXT X1
280 : NEXT Y1
290 :PRINT "FIN";:GOTO 700
300 :
310 :IF XC=X(3) THEN IF YC=Y(3) THEN 700
312 :X(P)=XC:Y(P)=YC
315 :CURSET XC,YC,0:CURSET XC,YC,1
320 NEXT P
330 :
690 REM-------------Rotation---
700 A=PI/2
705 FOR I=3 TO P-1
707 :X=X(I)-X0:Y=Y(I)-Y0
710 :S=SIN(A):IF S>-.01 AND S<.01 THEN S=0
720 :C=COS(A):IF C>-.01 AND C<.01 THEN C=0
730 :X1=X*C+Y*S
740 :Y1=Y*C-X*S
750 :CURSET 60+X1,60+Y1*1.0,1
760 NEXT I
```

ANIMATION

Robot

Ce programme transforme un dessin en DATA sous forme de caractères semi-graphiques. Ces caractères semi-graphiques sont affichés très rapidement, permettant ainsi d'animer des figures.




```
30 DIM LG$(30)
40 :
50 :
100 DATA "  *******
110 DATA "  *     *
120 DATA " ** * * **
130 DATA " **  *  **
140 DATA "  *     *
150 DATA "  * *** *
160 DATA "  *     *
170 DATA "* *******
180 DATA "*    *
190 DATA " *********
200 DATA "  *     * *
210 DATA "  *     *  *
220 DATA "  *     *
230 DATA "  *     *
240 DATA "  *     *
250 DATA "  *     *
260 DATA "  *     *
270 DATA "  *     *
280 DATA "  *******
290 DATA "   *   *
300 DATA "   *   *
310 DATA "   *   *
320 DATA "   *   *
330 DATA "  **   **
340 :
350 :
1000 FOR I=1 TO 24
1010 :READ LG$(I)
1020 NEXT I
1030 REM--------------
1100 FOR PV=1 TO 3              :REM 3 RANGEES
1105 : FOR P=1 TO 2             :REM 2 BLOCS PAR RANGEE
1110 : FOR L=1 TO 8             :REM 8 LIGNES PAR RANGEE
1115 :  L1=(PV-1)*8+L:PS=(P-1)*6+1
1120 :  X$(L)=MID$(LG$(L1),PS,6)
1130 : NEXT L
1135 : C=64+(PV-1)*2+P        :REM 'A' 1ER CARACTERE
1136 : D=C*8                  :REM DEPLACEMENT MEMOIRE ECRAN
1140 : GOSUB 2000
1200 : NEXT P
1300 :
1500 NEXT PV
```

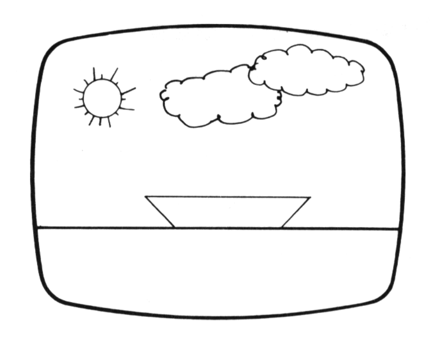
```
1502 REM---------------- ESSAIS
1505 CLS
1507 FOR X=1 TO 30  STEP 5
1510 :PLOT X,10,"AB"
1520 :PLOT X,11,"CD"
1530 :PLOT X,12,"EF"
1540 NEXT X
1545 WAIT 300
1549 :
1550 :
1551 :
1552 :
1553 REM-------------- DEPLACEMENT
1555 CLS
1600 FOR X=1 TO 30
1610 : PLOT X,10,"AB"
1620 : PLOT X,11,"CD"
1630 : PLOT X,12,"EF"
1640 :
1645 : WAIT 5
1650 : PLOT X,10,"  "
1660 : PLOT X,11,"  "
1670 : PLOT X,12,"  "
1680 NEXT X
1700 STOP
1990 REM--------------- CV BINAIRE--> DECIMAL
2000 A=46080
2010 FOR N=1 TO 8
2015 :ND=0
2020 : L=LEN(X$(N))
2025 : FOR M=1 TO 7
2030 :  X$=MID$(X$(N),M,1)
2035 :  IF X$="*" THEN R=1 ELSE R=0
2040 :  ND=ND+R*(2^(8-M-2))
2050 : NEXT M
2060 : POKE (A+D+N-1),ND
2070 NEXT N
2080 RETURN
```

```
10 REM      METHODE DIRECTE
20 :
30 DATA 7,4,13,12,4,5,4,23   :REM redefinition de A
40 DATA 60,4,22,38,4,52,4,60 :REM redefinition de B
50 :
60 A=46080              :REM adresse memoire ecran
70 FOR C=0 TO 1         :REM 2 caracteres
80 : D=(ASC("A")+C)*8:REM A Premier caractere
90 : FOR N=0 TO 7
100 :   READ ND:POKE A+D+N,ND
110 : NEXT N
120 :NEXT C
130 PLOT X,10,"AB"
140 REM les valeurs des codes en DATA sont
150 REM obtenues en ajoutant au Programme Precedant
170 REM   2065  PRINT ND
180 REM   2075  PRINT
```

PAYSAGE

Paysage

Soleil, nuages, mer et le bateau avance!...



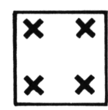
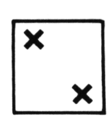
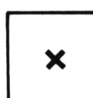
```
5 REM      PAYSAGE
6 :
10 HIRES:PAPER 7:INK 4
15 :
16 :
18 REM------------------------- NUAGES
20 FOR NU=1 TO 3
22 :X0=(NU-1)*70+40:Y0=50+RND(1)*10
25 :FOR N=1 TO 10
30 :X=RND(1)*40+X0
40 :Y=RND(1)*20+Y0
50 :
60 :R=5+RND(1)*10
70 :
90 :CURSET X,Y,1
100 :FOR R1=1 TO R:CIRCLE R1,1:NEXT R1
120 :NEXT N
130 :
140 NEXT NU
150 REM--------------------- SOLEIL
200 :
210 CURSET 1,150,1
220 FILL 49,1,22
230 REM---
250 FOR AM=40960 TO 40960+37*40 STEP 40
260 :POKE AM+1,3
265 :POKE AM+8,4
270 NEXT AM
280 CURSET 25,20,1
290 FOR R=1 TO 10:CIRCLE R,1:NEXT R
291 :
292 FOR A=0 TO 2*PI STEP PI/10
294 :R=10+RND(1)*8
295 :X=R*COS(A):Y=R*SIN(A)
296 :CURSET 25,20,1:DRAW X,Y,1
298 NEXT A
300 REM------------------------ BATEAU
310 FOR Y=1 TO 6
320 :CURSET 50+Y,144+Y,1
330 :DRAW 40-2*Y,0,1
340 NEXT Y
350 REM-------- BATEAU MOBILE
400 FOR X= 50 TO 200
410 : FOR Y=1 TO 6
420 : CURSET X+Y,144+Y,0
430 : CURSET X+40-1*Y,144+Y,1
440 : NEXT Y
450 NEXT X
```

DESSIN DE DÉS

Dessins de dés

Ce programme dessine des tires au hasard.

```
10 REM       Dessin de des
14 :
15 REM       Dessine 3 des tires au hasard
20 :
90 LORES 0
100 FOR TR=1 TO 3                3 Des
110 :DE=INT(RND(1)*6)+1
190 :X=TR*10:Y=10
192 :
195 : FOR X1=X-2 TO X+4
196 :  PLOT X1,Y-2,17:PLOT X1,Y+4,17
197 : NEXT X1
200 : FOR Y1=Y-2 TO Y+4
201 :  PLOT X-2,Y1,17:PLOT X+4,Y1,17
202 : NEXT Y1
203 :
204 :IF DE=1 OR DE=3 OR DE=5 THEN PLOT X+1,Y+1,"X"
205 :IFDE=>2 THEN PLOT X,Y,"X":PLOT X+2,Y+2,"X"
210 :IF DE=>4 THEN PLOT X,Y,"X":PLOT X+2,Y,"X":PLOT X,Y+2,"X":PLOT
X+2,Y+2,"X"
220 :IF DE=6 THEN PLOT X+1,Y,"X":PLOT X+1,Y+2,"X"
290 :
300 NEXT TR
```

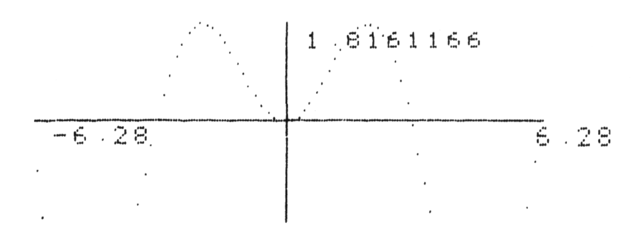
TRACÉ DE COURBE

Tracé de courbe avec graduation

```
10 REM TRACE DE COURBE (avec graduation)
20 :
30 HECR=150:LECR=200       :REM Hauteur/longueur ecran
40 INPUT "BORNE X1 ";B1
50 INPUT "BORNE X2 ";B2
60 INPUT "PAS      ";PAS
70 :
75 X=B1:GOSUB 290:Y1=Y:Y2=Y
80 FOR X=B1 TO B2 STEP PAS
100 : GOSUB 290
110 :  IF Y<Y1 THEN Y1=Y      :REM Mini
120 :  IF Y>Y2 THEN Y2=Y      :REM Maxi
130 NEXT X
135 :
140 HIRES
150 EX=LECR/(B2-B1)          :REM Echelle x
160 EY=(HECR-2)/(Y2-Y1)      :REM Echelle Y
170 REM----------------------------- Axes ---
180 IF Y2=>0 THEN IF Y1<=0 THEN Y=HECR-ABS(Y1)*EY:CURSET 1,Y,1:DRAW
 LECR,0,1
190 IF B2=>0 THEN IF B1<=0 THEN X=ABS(B1)*EX:CURSET X,1,1:DRAW 0,HE
CR,1
200 REM------
210 FOR X=B1 TO B2 STEP PAS
220 :GOSUB 290
230 :SX=(X-B1)*EX
240 :SY=HECR-(Y-Y1)*EY
250 :CURSET SX,SY,1
260 NEXT X
261 REM------------------- Graduations
262 Y=HECR-ABS(Y1)*EY+3:X=1:B=B1:GOSUB 273
264 Y=HECR-ABS(Y1)*EY+3:X=LECR-9:B=B2:GOSUB 273
266 Y=HECR+4:X=ABS(B1)*EX:B=Y1:GOSUB 273
269 Y=5      :X=ABS(B1)*EX:B=Y2:GOSUB 273
271 GOTO 284
272 REM----
273 IF X>LECR+10 THEN X=LECR/2
274 IF Y>HECR+10 THEN Y=HECR/2
275 CURSET X,Y,3::X$=STR$(B):IF ASC(X$)=2 THEN X$=RIGHT$(X$,LEN(X$)
-1)
276 FOR P=1 TO LEN(X$):C=ASC(MID$(X$,P,1)):CURSET X+P*8,Y,3:CHAR C,
0,1
277 NEXT P
278 RETURN
280 :
284 GET X$
286 TEXT
287 END
288 REM --------------------- Courbe a tracer ----
290 Y=SIN(X)*X
300 RETURN
```

Copie écran → imprimante SEIKOSHA (haute résolution)

```
10 REM   COPIE ECRAN-->IMPRIMANTE SEIKOSHA (HAUTE RESOLUTION)
20 :
30 DIM T(240):POKE 49,255:CALL #E6CA:LPRINT :Y=0
40 FOR N=1 TO 28
50 :GOSUB 200
60 : FOR M=1 TO 7
70 :  GOSUB 300
80 : NEXT M
90 : GOSUB 400
100 NEXT N
110 GOSUB 200
120 :
130 FOR M=1 TO 4
140 : GOSUB 300
150 NEXT M
160 GOSUB 400:CALL #E804:STOP
170 :
200 REM------ Remise a zero -----------
210 FOR I=1 TO 240:T(I)=128:NEXT I:RETURN
300 REM------ Traitement par ligne ----------
310 FOR X=0 TO 239
320 :P=POINT(X,Y):IF P=-1 THEN T(X)=T(X)+2^(M-1)
330 NEXT X
340 Y=Y+1
350 RETURN
400 REM---------- Edition-----
410 LPRINT CHR$(#8);
420 FOR I=1 TO 240
430 :LPRINT CHR$(T(I));
440 NEXT I
450 LPRINT:RETURN
```



Conduite voiture.

```
10 REM     CONDUITE VOITURE
20 :
30 REM---------- GENERATION VOITURE
35 DATA 45,63,45,12,12,45,63,45
37 V$="!"
38 A=46080
39 D=ASC(V$)*8
40 FOR N=0 TO 7
42 :READ ND
43 :POKE A+D+N,ND
45 NEXT N
46 REM-------------- ROUTE
50 L$=CHR$(255)+"     .      "+CHR$(255)
55 REM------
60 CLS:PAPER7:INK4
70 XX=12
80 XV=XX+5:YV=2             :REM Position voiture
100 FOR KM=1 TO 30000
110 :X=INT(RND(1)*3)-1
120 :XX=XX+X
125 :IF XX>26 THEN XX=XX-1
126 :IF XX<3 THEN XX=XX+1
130 :PRINT SPC(XX);L$
132 :IF SCRN(XV,YV)<>127 THEN 140
135 :EXPLODE:PAPER1:WAIT 30:PAPER 5:WAIT 30:PAPER 7
136 :PRINT KM/100;" KM":WAIT 200:GOTO 60
140 :
142 :PLOT XV,YV,V$
145 :PLOT XV,YV-1," "
150 :X$=KEY$:IF X$="" THEN 160
155 :IF ASC(X$)=9 THEN XV=XV+2
156 :IF ASC(X$)=8 THEN XV=XV-2
160 :
200 NEXT KM
500 FOR Y=0 TO 27
505 FOR X=0 TO 38
510 IF SCRN(X,Y)=127 THEN LPRINT "X";:GOTO 600
520 LPRINT CHR$(SCRN(X,Y));
600 NEXT X
605 LPRINT
610 NEXT Y
990 REM-------------- POUR VOITURE VERTE
1000 :
1010 :  140 PLOT XV-1,YV,2
1020 :  142 PLOT XV,YC,V$
1030 :  143 PLOT XV+1,YV,4
```

```
X    .     X
 X          X
 X    .     X
 X    .     X
 X    .     X
  X    .     X
   X    .     X
    X    .     X
     X    .     X
     X    .     X
    X    .     X
   X    .     X
   X    .     X
   X    .     X
   X    .     X
```

Pièges

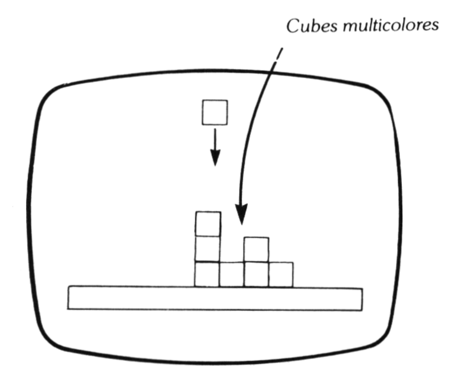
```
10 REM      PIEGES
12 :
13 REM   3 monstres vous poursuivent
14 REM   vous cherchez a les faire tomber
16 REM   dans des pieges
17 :
19 PRINT CHR$(6);
20 :
30 LORES 0
40 V=1                  ' vitesse monstres
45 M=0                  ' monstres morts
50 XC=18:YC=10:PLOT XC,YC,17   ' vos coordonees
60 :
70 FOR I=1 TO 3
80 :XM(I)=INT(RND(1)*38):IF ABS(XM(I)-XC)<8  THEN 80
90 :YM(I)=INT(RND(1)*23):IF ABS(YM(I)-YC)<8 THEN 90
95 :PLOT XM(I),YM(I),19                    ' monstres jaunes
100 :PLOT RND(1)*XC+10,RND(1)*YC+7,20   ' pieges bleus
110 NEXT I
115 WAIT 200
120 REM------
130 FOR I=1 TO 3
140 : X=PEEK(#208):IF X=56 THEN 220 :REM test clavier
150 : PLOT XC,YC,16
160 :
170 : IF X=172 THEN IF XC>0 THEN XC=XC-1
180 : IF X=188 THEN IF XC<37 THEN XC=XC+1
190 : IF X=180  THEN IF YC<23 THEN YC=YC+1
200 : IF X=156 THEN IF YC>0 THEN YC=YC-1
210 : PLOT XC,YC,17
215 :
220 : IF XM(I)=0 THEN 400
230 : XA=XM(I):YA=YM(I)
240 : XM(I)=XM(I)-(SGN(XM(I)-XC)*V)
250 : YM(I)=YM(I)-(SGN(YM(I)-YC)*V)
254 : REM --- tests pieges
255 : IF SCRN(XM(I),YM(I))<>20 THEN 275
256 :
260 : XM(I)=0:YM(I)=0:PLOT XA,YA,16:M=M+1:GOTO 300
270 :REM--- test devore
275 : X1=INT(XM(I)):Y1=INT(YM(I))
280 : PLOT XA,YA,16:PLOT X1,Y1,19
285 :
290 : IF X1=XC AND Y1=YC THEN PLOT 2,2,"PERDU":WAIT 300:GOTO 20
295 :
300 : IF M=3 THEN PLOT 2,2,"GAGNE":WAIT 300:GOTO 20
400 NEXT I
410 GOTO 130
```

Monstres

Vous

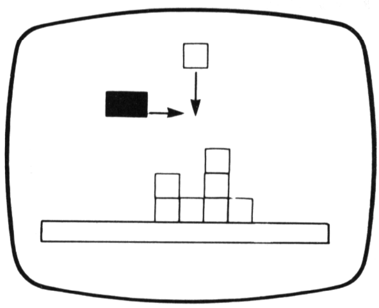
Pièges

Tas de cubes



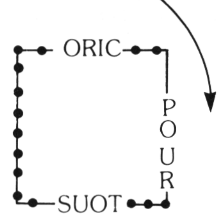

```
10 REM   TAS DE CUBES
20 :
30 REM    Des cubes multicolores tombent du haut de l'ecran
32 REM   rebondissent et forment un tas multicolore
34 :
40 LORES0
50 FOR X=1 TO 38:PLOT X,24,17:NEXT X
60 :
70 X=20
80 C=INT(RND(1)*7)+1
90 WAIT 20
100 FOR Y=1 TO 24
105 :IF SCRN(X,Y)<>16 THEN 120
110 :IF SCRN(X,Y)=16 THEN PLOT X,Y-1,16:PLOT X,Y,16+C
115 NEXT Y
119 :
120 :SHOOT
130 : D=INT(RND(1)*2):IF D=0 THEN D=-1
140 :FOR R=1 TO 24
150 : IF SCRN(X,Y)=16 THEN PLOT X,Y-1,16:PLOT X,Y,16+C:Y=Y+1:GOTO 1
50
160 : IF SCRN(X+D,Y)<>16 THEN 70
170 : PLOT X,Y-1,16 :X=X+D:PLOT X,Y,16+C
180 : PING:WAIT 10
190 :Y=Y+1
210 :NEXT R
220 GOTO 70
```

Soucoupe



```
10 REM    SOUCOUPE
20 :
25 REM       Le joueur doit rattraper des paves
26 REM       tombant du haut de l'ecran
30 :
40 LORES0
42 PRINT CHR$(17)      :REM suppression curseur
45 :
50 YS=14:XS=0          :REM soucoupe
52 PLOT 19,4,"APPUYER SUR <ESPACE>"
60 N=N+1
70 DP=0
80 FOR X=XS TO XS+2:PLOT X,YS,17:NEXT X
85 :
90 XP=5 +INT(RND(1)*7):YP=0:C=16+2+INT(RND(1)*6)
100 PLOT XP,YP,C
110 WAIT 150
120 REM----------------- depart soucoupe
130 X$=KEY$:IF X$<>"" THEN DP=1
140 :
150 WAIT 3
155 IF SCRN(XP,YP+1)<18 AND YP<23 THEN 170
160 PLOT XS,YS,16:PLOT XS+1,YS,16:PLOT XS+2,YS,16:GOTO 50
165 :
170 IF YP<23 THEN YP=YP+1
180 :
190 IF DP=0 THEN 320
200 IF XS>22 THEN PLOT XS,YS,16:PLOT XS+1,YS,16:PLOT XS+2,YS,16:GOT
O 50
210 XS=XS+1
220 :
230 IF SCRN(XP,YP)<>17 THEN 300
240 REM---- gagne
250 R=R+1:X=INT(R/N*100):PLOT 28,2,"SCORE:"+STR$(X)+"% "
260 WAIT 100
270 PLOT XP,YP-1,16
280 PLOT XS-1,YS,16:PLOT XS,YS,16:PLOT XS+1,YS,16
290 GOTO 50
295 :
300 REM------ Affichage nouvelles positions
310 PLOT XS-1,YS,16:PLOT XS+2,YS,17
315 :
320 PLOT XP,YP-1,16:PLOT XP,YP,C
340 GOTO 130
```

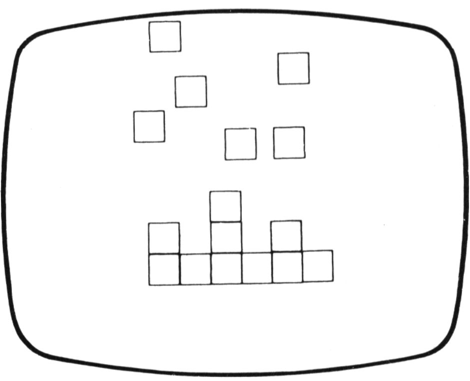
Affichage publicitaire




```
1 REM  AFFICHAGE PUBLICITAIRE
2 :
3 REM   Affiche un message sur le perimetre d'un rectangle
4 :
5 :
6 CLS:INK1
7 L=10:H=5:X0=5 :Y0=5      ' Longueur/hauteur
10 REM------------- Calcul coordonnees affichage
11 DIM X(100),Y(100)
12 FOR I=1 TO L
13 :X(I)=X0+I:Y(I)=Y0
14 NEXT I
15 FOR I=1 TO H
16 :X(L+I)=X0+L:Y(I+L)=Y0+I
17 NEXT I
19 :
20 FOR I=1 TO L
21 : X(L+H+I)=X0+L-I:Y(L+H+I)=Y0+H
22 NEXT I
23 :
24 FOR I=1 TO H
25 :X(L+H+L+I)=X0:Y(L+H+L+I)=Y0+H-I
26 NEXT I
28 REM---------------
30 A$="....ORIC..POUR..TOUS........................."
35 LG=L*2+H*2+1
40 X$=LEFT$(A$,LG):L=LEN(X$)
45 :
50 :
60 X$=RIGHT$(X$,1)+LEFT$(X$,L-1)
65 :
70 :
100 FOR I=1 TO L
110 PLOT X(I),Y(I),MID$(X$,I,1)
120 NEXT I
130 GOTO 60
```

Cubes multicolores

```
10 REM  CUBES MULTICOLORES
20 :
21 REM   Des cubes multicolores tombent du haut
22 REM   de l'ecran et s'empilent
24 :
30 LORES0
32 PRINT CHR$(17)          ' Annulation curseur/ ctrl Q
35 :
37 N=10:DIM X(N+1),Y(N+1)   ' Nombre de cubes
38 REM----------- Coordonnees cubes de depart
40 FOR F=1 TO N
50 :X=INT(RND(1)*10):Y=INT(RND(1)*24):C=RND(1)*7+1+16:PLOT X,Y,C
60 :PLOT X,Y,C
65 :X(F)=X:Y(F)=Y:C(F)=C
70 NEXT F
80 REM------------- Descente cubes
100 FOR F=N TO 1 STEP-1
115 :IF Y(F)>23 THEN Y(F)=0:X(F)=INT(RND(1)*10):GOTO 150
116 :IF SCRN(X(F),Y(F)+1)<>16 THEN Y(F)=0:X(F)=INT(RND(1)*10):GOTO
150
120 :PLOT X(F),Y(F),16:PLOT X(F),Y(F)+1,C(F)
130 :Y(F)=Y(F)+1
140 :
150 NEXT F
160 :
230 :
300 GOTO 100
```



Maison

```
10 REM      MAISON
20 :
30 :
35 :
45 CF=7                   :REM couleur de fond
50 HIRES:PAPER CF:INK1
60 REM------------------ Toit rouge
80 X0=100:Y0=100
85 HT=30:L=60             :REM hauteur toit/longueur
90 FOR I=1 TO HT-1
92 :CURSET X0-I/2,Y0+I,1
94 :DRAW L +I,0,1
96 NEXT I
100 REM------------- Cheminee rouge
102 FOR I=1 TO 10
104 :CURSET X0+15,Y0-10+I,1
105 :DRAW 5,0,1
107 NEXT I
110 REM---------------- Maison bleu fonce
120 X1=X0:Y1=Y0+HT:H=50:C=4:L=65:GOSUB 500
200 REM---------------- fenetre jaune
202 FOR I=1 TO 15
203 :CURSET X0+10,Y0+HT+15+I,1
204 :DRAW 15,0,1
205 NEXT I
210 :
220 CURSET X0+10+7,Y0+HT+15,1:DRAW 0,15,0
230 CURSET X0+10,Y0+HT+15+15/2,1:DRAW 15,0,0
240 REM-------------------- Porte noire
250 CURSET X0+40,Y0+HT+15,1
255 FILL 35,1,16+0
257 CURSET X0+55,Y0+HT+15,1
258 FILL 35,1,16+4
270 CURSET X0+45,Y0+HT+30,1     :REM Poignee
280 :
285 REM--------- Changement couleur de fond (optionnel)
290 WAIT 200
295 REM----------------- Bleu
296 CURSET (X1+L)+1,Y1,1:FILL H+1,1,16+6:PAPER 6:WAIT 100
305 CURSET (X1+L)+1,Y1,1:FILL H+1,1,16+0:PAPER 0:WAIT 100
320 CURSET (X1+L)+1,Y1,1:FILL H+1,1,16+CF:PAPER CF
400 GOTO 290
490 REM----------------- Foreground/background
500 CURSET X1,Y1,2:FILL H,1,16+4       :REM background bleu
502 CURSET X1+L,Y1,2:FILL H,1,16+CF
504 :
506 CURSET X1-12,Y1,2:FILL H,1,3       :REM foreground jaune
508 CURSET X1+L-12,Y1,2:FILL H,1,7
510 RETURN
```

Soucoupe volante

```
10 REM      SOUCOUPE VOLANTE
20 :
30 REM Une soucoupe laisse tomber 1 ou plusieurs
35 REM projectiles (en appuyant sur <ESPACE>
37 PRINT CHR$(17)
40 REM--------------- Definition soucoupe
50 DATA 12,12,30,63,63,63,63,0
60 SC$="!":A=46080:D=ASC(SC$)*8
80 FOR N=0 TO 7:READ ND:POKE A+D+N,ND:NEXT N
101 REM-------------- Definition projectile
102 DATA 0,0,0,0,0,0,0,6
103 P$="@":D=ASC(P$)*8
104 FOR N=0 TO 7:READ ND:POKE A+D+N,ND:NEXT N
107 C$=CHR$(127)
110 REM--------------------
120 PLAY 1,0,0,0:CLS:PAPER7:INK1
122 :
127 PLOT 1,24,"Appuyer sur <ESPACE>"
130 :
135 XC=INT(RND(1)*20)+10:YC=21:PLOT XC,YC,C$:PLOT XC-1,YC,4
140 YS=2
145 X1=1:X2=37:S=1                         :REM bornes ecran/sens
150 :
155 FOR I=1 TO 5:YP(I)=0:XP(I)=0:NEXT I
157 :
160 FOR D=X1 TO X2 STEP S
170 :XS=D
174 :PLOT XS-S,YS,1:PLOT XS,YS,SC$    :REM soucoupe rouge
178 :
180 :IF KEY$="" THEN 185
181 :FOR I=1 TO 5
182 :IF YP(I)=0 THEN YP(I)=YS+1:XP(I)=XS:GOTO 185
183 :NEXT I
184 :
185 :FOR I=1 TO 5
190 : IF YP(I)=0 THEN 220
193 : IF YP(I)<>YC OR XP(I)<>XC THEN 200
195 : EXPLODE:PLOT XC,YC,C$:PLOT XC-1,YC,5:WAIT 30:PLOT XC-1,YC,1:W
AIT 400
196 : GOTO 120
197 :
200 : PLOT XP(I),YP(I)," "
210 : IF YP(I)<23 THEN YP(I)=YP(I)+1:PLOT XP(I),YP(I),P$
220 :NEXT I
290 :SOUND 1,D*3,12
300 NEXT D
310 PLOT XS,YS," ":X=X1:X1=X2:X2=X:S=-S     :REM inversion sens
330 YS=YS+2:IF YS>10 THEN GOTO 120
340 GOTO 160
```

Soucoupe rouge

Cible bleue

```
10 REM  IMMEUBLES DE COULEURS
20 :
30 REM  dessine des immeubles de couleur aleatoire
35 REM
40 CLS
50 REM---------- redefinition graphique de '!'
60 DATA 63,63,51,51,51,63,63,63
70 :
80 A=46080:D=ASC("!")*8
90 :
100 FOR N=0 TO 7
110 :READ ND:POKE A+D+N,ND
120 NEXT N
130 REM-----------------------------
180 X0=1 :Y0=23
185 XB=X0:YB=Y0
188 :
190 FOR N=1 TO 5       :REM 5 immeubles
200 :H=RND(1)*6+5
210 :L=RND(1)*4+3
220 :C=RND(1)*7
230 :GOSUB 500
235 :XB=XB+L+1
240 NEXT N
480 STOP
490 REM------------------ Dessin immeuble ---
500 FOR Y=YB TO YB-H STEP-1
505 :AF$=CHR$(C)
510 :FOR X=1 TO L
530 : AF$=AF$+"!"
540 :NEXT X
550 :PLOT XB,Y   ,AF$
560 NEXT Y
570 RETURN
580 :
```

ROUGE

BLEU

VERT

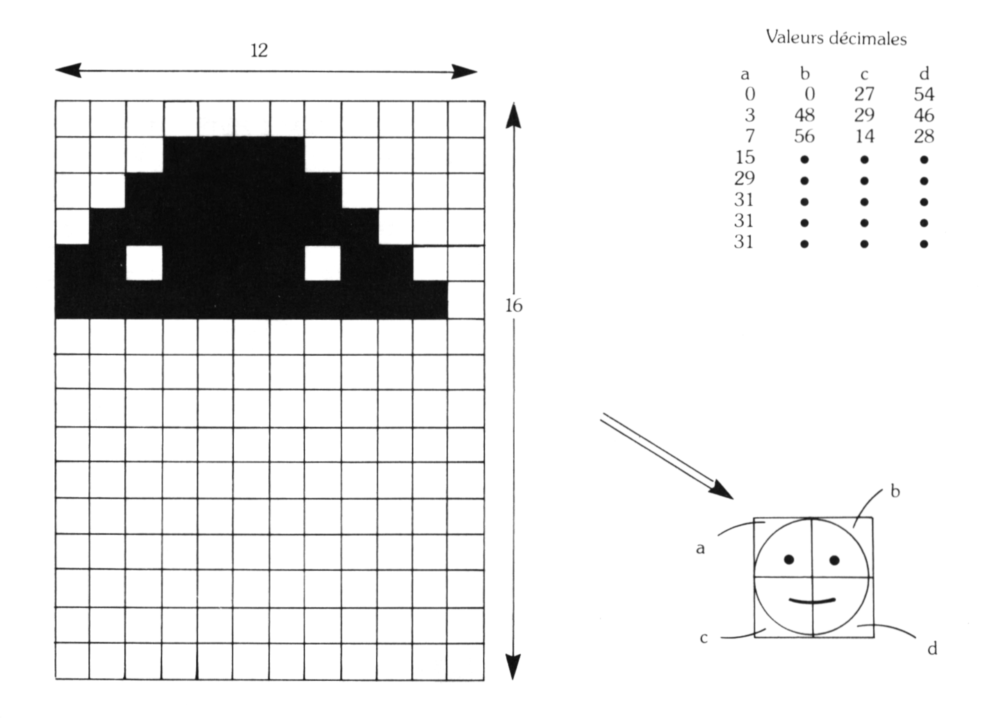
Caractère redéfini.

GENERATION DE CARACTERES

Nous reprenons le programme 'ROBOT' où un dessin en 'DATA' est reproduit à l'aide de caractères graphiques. Mais cette fois, le dessin d'origine est réalisé directement à l'écran suivant le principe du 'TELECRAN'.

- 'L' permet de déplacer le curseur sans dessiner (ou d'effacer)
- 'F' termine le dessin.

La deuxième partie du programme calcule les valeurs décimales nécessaires à la définition des caractères choisis (a, b, c, d, sur l'exemple).



Valeurs décimales

a	b	c	d
0	0	27	54
3	48	29	46
7	56	14	28
15	•	•	•
29	•	•	•
31	•	•	•
31	•	•	•
31	•	•	•

Nous avons prévu des dessins pouvant comporter jusqu'à 16 x 12 points.
Lorsque le dessin est affiché sous forme graphique, nous avons la faculté de modifier le dessin d'origine et d'obtenir un nouveau dessin graphique ainsi que de nouvelles valeurs décimales.

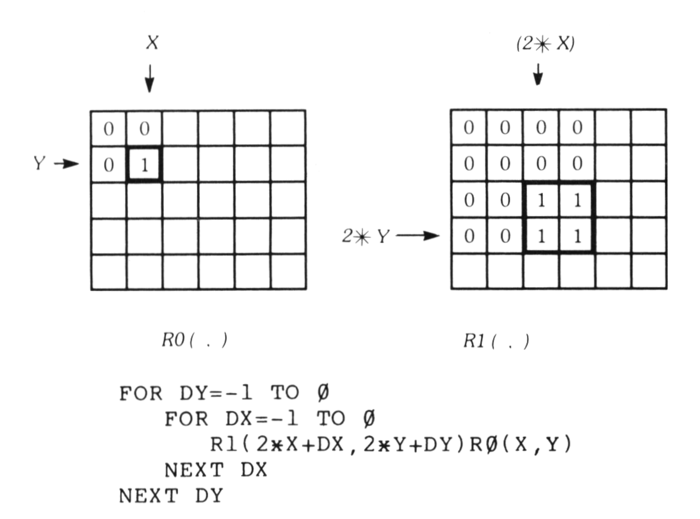
Le programme ci-dessous utilise les valeurs décimales obtenues pour redéfinir les caractères a, b, c, d.

```
10 DATA 0,3,7,15,29,31,31,31
20 DATA 0,48,56,60,46,62,62,62
30 DATA 27,29,14,7,3,0,0,0
40 DATA 54,46,28,56,48,0,0,0
60 A=46080
65 :
70 FOR CA=1 TO 4          :REM 4caracteres
80  :C=ASC("a")-1+CA:D=C*8
90  :FOR N=0 TO 7
100  :  READ ND:POKE A+D+N,ND
110  :NEXT N
120 NEXT CA
130 :
140 PLOT 10,10,"ab"
150 PLOT 10,11,"cd"
```

Extensions :

- Nous avons fait 'pivoter' le dessin obtenu (voir rotation de figures).
  Deux problèmes se posent :
  1) Les points de l'écran ont une hauteur différente de la largeur.
     Par conséquent, après une rotation de 90°, une figure se trouve 'aplatie'.
  2) Pour des rotations inférieures à 90°, les dessins ne sont pas précis.

- Nous pourrions générer des figures agrandies. Les figures obtenues permettraient de simuler des grossissements.




```
FOR DY=-1 TO Ø
   FOR DX=-1 TO Ø
      R1(2*X+DX,2*Y+DY)RØ(X,Y)
   NEXT DX
NEXT DY
```

Sur l'exemple, la taille est multipliée par 2. Un point de la figure d'origine est remplacé par 4 points.

Génération de caractères

```
10 REM        GENERATION DE CARACTERES
12 REM        (12X16 Points)
20 :
25 DIM R0(13,17),R1(13,17)
30 LORES0
31 REM------------------ cadre
32 FOR X=0 TO 13:PLOT X,0,18:PLOT X,17,18:NEXT X
33 FOR Y=0 TO 17:PLOT 0,Y,18:PLOT 13,Y,18:NEXT Y
34 REM--------------------- Saisie des Points
35 X=1:Y=1
36 PLOT 10,23,"L=LEVER / B=BAISSER /F=FIN"
40 :
50 C$=KEY$:IF C$<>"" THEN 80
60 PLOT X,Y,17:PLOT X,Y,16:GOTO 50
70 :
80 IF L=0 THEN PLOT X,Y,17
100 C=ASC(C$)
110 IF C=8 THEN IF X>1 THEN X=X-1
120 IF C=9 THEN IF X<12 THEN X=X+1
130 IF C=10 THEN IF Y<16 THEN Y=Y+1
140 IF C=11 THEN IF Y>1 THEN Y=Y-1
160 IF C$="L" THEN L=1
170 IF C$="B" THEN L=0
175 IF C$="F" THEN 200
180 GOTO 50
190 REM-----------------------------
200 FOR Y=1 TO 16
210 :FOR X=1 TO 12
230 :    IF SCRN(X,Y)=17 THEN R0(X,Y)=1ELSE R0(X,Y)=0
240 : NEXT X
250 NEXT Y
```

```
420 REM---------------
600 A=46080
610 FOR LG=1 TO 2        :REM 2 lignes
620 : FOR CL=1 TO 2      :REM 2 colonnes
625 : C=ASC("a")-1+(LG-1)*2+CL
627 : D=C*8
630 : FOR L=1 TO 8
635 :  L1=(LG-1)*8+L:PS=(CL-1)*6
637 :  ND=0
640 :  FOR X=1 TO 6
650 :   ND=ND+R0(PS+X,L1)*2^(6-X)
660 :  NEXT X
670 :  POKE (A+D+L-1),ND
675 :  PLOT 18+(LG-1)*6+(CL-1)*3,5+L,STR$(ND)+" "
680 : NEXT L
690 : NEXT CL
700 NEXT LG
910 :
920 PLOT 20,20,"ab"
930 PLOT 20,21,"cd"
932 :
933 :
934 :
935 :
936 :
937 :
938 :
939 :
940 :
941 :
942 :
943 :
944 :
950 REM=============== RETOURNEMENT (OPTIONNEL)
960 PLOT 20,18,"RETOURNEMENT (O/N)"
965 GET R$
970 IF R$<>"O" THEN 35
980 :
1000 FOR L=1 TO 16
1010 :FOR C=1 TO 12
1020 :R1(C,L)=R0(12-C+1,L)
1030 :NEXT C
1040 NEXT L
1050 :
1100 FOR L=1 TO 16
1110 :FOR C=1 TO 12
1120 :R0(C,L)=R1(C,L)
1130 :NEXT C
1140 NEXT L
1150 :
1160 GOTO 600
```

Ciel étoilé multicolore

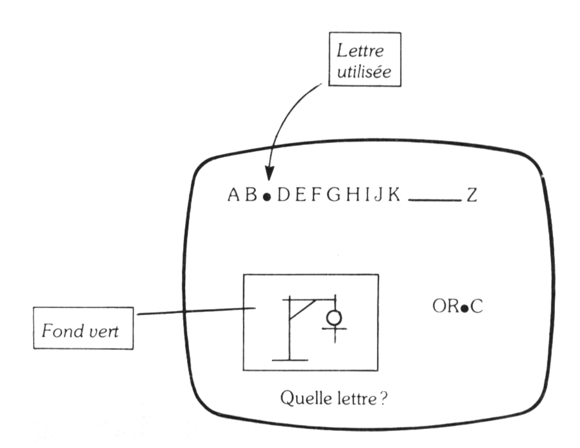
```
5 REM  CIEL ETOILE MULTICOLORE
6 :
7 :
8 REM   Dessine des etoiles multicolores
9 :
10 :
15 DATA 1,1,3,3,63,31,15,15
20 DATA 0,0,32,32,62,60,56,56
30 DATA 30,24,48,32,0,0,0,0
40 DATA 60,12,6,2,0,0,0,0
60 A=46080
65 :
70 FOR CA=1 TO 4            :REM 4caracteres
80 :C=ASC("a")-1+CA:D=C*8
90 :FOR N=0 TO 7
100 : READ ND:POKE A+D+N,ND
110 :NEXT N
120 NEXT CA
130 :
140 PLOT 10,10,"ab"
150 PLOT 10,11,"cd"
160 PAPER0:CLS
165 PRINT CHR$(17);
190 REM-------------
200 FOR E=1 TO 50
210 :X=RND(1)*35+1
220 :Y=RND(1)*23
221 :IF SCRN(X-1,Y)<>32 THEN 210
222 :IF SCRN(X-1,Y+1)<>32 THEN 210
224 :C=RND(1)*7+1
225 :PLOT X-1,Y,C:PLOT X-1,Y+1,C
230 :PLOT X,Y,"ab"
240 :PLOT X,Y+1,"cd"
300 NEXT E
305 WAIT 300
310 :
330 PAPER6:WAIT 200
340 PAPER7:WAIT 200
350 PAPER0:WAIT 200
360 GOTO 330
```

Saut obstacles

```
5 REM      SAUT OBSTACLES
6 :
7 REM      Appuyer sur <ESPACE> pour sauter les obstacles
8 :
9 REM------------------------ DESSIN1
10 DATA 3,3,3,1,7,13,1,1
20 DATA 32,32,32,0,0,48,24,0
30 DATA 3,2,6,4,12,8,12,0
40 DATA 32,48,24,8,8,8,12,0
41 REM------------------------ DESSIN SAUT
42 DATA 3,3,3,1,15,1,1,1
43 DATA 32,32,32,0,0,60,0,0
44 DATA 3,2,6,28,16,0,0,0
45 DATA 32,48,24,12,6,0,0,0
60 A=46080
65 REM----------------- REDEFINITION A....H
70 FOR CA=1 TO 8
80 :C=ASC("a")-1+CA:D=C*8
90 :FOR N=0 TO 7
100 : READ ND:POKE A+D+N,ND
110 :NEXT N
120 NEXT CA
130 :
150 CLS:INK4:PAPER7
155 Y=10
157 FOR X=0 TO 38:PLOT X,Y+2,CHR$(255):NEXT X
158 REM=====================
160 PLOT 15,Y+1,"!"
170 PLOT 25,Y+1,"!"
180 :
190 REM--
205 FOR X=2 TO 36 STEP1
206 :IF SCRN(X+1,Y+1)<>33 THEN 209
207 :SHOOT:PD=PD+1:PLOT20,20,STR$(PD)+" OBSTACLE(S)":GOTO 210
208 :
209 :IF SCRN(X,Y+1)=33 THEN SHOOT:PD=PD+1:PLOT 20,20,STR$(PD)+" OBS
TACLE(S)"
210 :PLOT X,Y,"ab"
220 :PLOT X,Y+1,"cd"
225 :WAIT 10
240 :PLOT X,Y,"  "
250 :PLOT X,Y+1,"  "
255 :
260 :IF KEY$<>"" THEN GOSUB 500
300 NEXT X
305 PS=PS+1:PLOT 20,22,STR$(PS)+" PASSAGE(S)"
320 GOTO 160
490 REM-------------- SAUT
500 :
505 PLOT X,Y-1,"ef":PLOT X,Y,"gh"
510 WAIT 20
520 PLOT X,Y-1,"  ":PLOT X,Y,"  "
525 X=X+2
530 RETURN
```

JEU DU PENDU

- Voici le classique **JEU DU PENDU**. Celui-ci présente cependant une particularité : le pendu agite les pieds avant d'expirer.



Les lettres de l'alphabet qui n'ont pas encore été utilisées sont affichées en haut de l'écran.

Avant d'afficher un caractère sur l'écran avec **CHAR code, JEU, COULEUR,** nous effaçons le caractère précédent avec **CHAR 127, 0, 0,**. (0 représente la couleur de fond). Sans cet effacement préalable, il y aurait superposition de l'ancien et du nouveau caractère.

Jeu du pendu

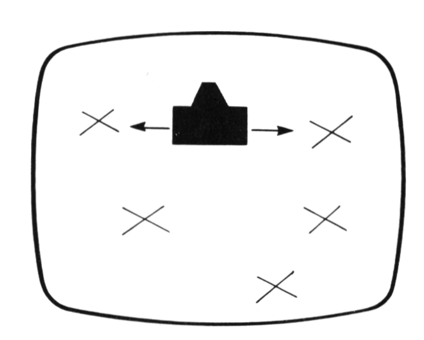
```
10 REM        JEU DU PENDU
15 :
20 REM  C.HIRCH
21 :
22 HIMEM #97FF
25 DIM MOT$(50)
26 FOR I=1 TO 50
27 :READ MOT$(I):IF MOT$(I)="*" THEN NM=I-1:GOTO 30
28 NEXT I
30 DIM AH$(26)
32 REM-----------
35 X0=120:Y0=100        :REM affichage mot
36 XA=20:YA=10          :REM affichage alphabet
38 REM------------------
50 FOR I=1 TO 26:AH$(I)=CHR$(I+64):NEXT I        :REM alphabet
55 HIRES:PAPER7:INK4
70 REM------------------- Fond vert pour pendu--
72 CURSET 0,95 ,3:FILL 98,1,16+2:CURSET 100,95 ,3:FILL 98,1,16+7
75 REM-------
110 CURSET XA,YA,3:FOR I=1 TO 26:CHAR ASC(AH$(I)),0,1:CURMOV 8,0,3:
NEXT I
130 :
140 X=RND(1)*NM+1:MOT$=MOT$(X):L=LEN(MOT$)
145 BPL=0:PD=0:LU$=""
160 CURSET X0,Y0,3:FOR P=1 TO L:CHAR 46,0,1:CURMOV 10,0,3:NEXT P
165 :
170 INPUT"Quelle lettre ";L$:PRINT:PRINT
180 IF LEN(LU$)=0 THEN 185
181 FOR P=1 TO LEN(LU$)              :REM lettre utilisee?
182 :IF L$=MID$(LU$,P,1) THEN GOSUB 900:GOTO 240
183 NEXT P
184 REM------- bonne lettre?
185 B=0
190 FOR P=1 TO L
200 :IF L$=MID$(MOT$,P,1) THEN B=1:LU$=LU$+L$:BPL=BPL+1:GOSUB 700:G
OSUB 715
210 NEXT P
220 IF B=0 THEN GOSUB 900:GOSUB 715
225 IF BPL=L THEN PRINT "BRAVO..":WAIT 300:GOTO 50
230 :
240 IF PD=8 THEN PRINT "PERDU.. Le mot etait:";MOT$:WAIT 300:GOTO 5
0
250 GOTO 170
```

```
690 REM-------------- affichage mot et alphabet
700 CURSET X0+10*(P-1),Y0,3:CHAR 127,0,0:CHAR ASC(L$),0,1:RETURN
715 X1=XA+(ASC(L$)-64-1)*8:CURSET X1,YA,3:CHAR 127,0,0:CHAR 46,0,1:
RETURN
720 :
721 :
722 :
723 :
724 :
725 :
726 :
727 :
895 REM------------- dessin pendu ---------
900 PD=PD+1:PING
910 ON PD GOSUB 920,930,940,950,960,970,980,984:RETURN
916 :
920 CURSET 20,190,3:DRAW 20,0,1:RETURN
930 CURSET 30,190,3:DRAW 0,-80,1:RETURN
940 CURSET 30,115,3:DRAW 40,0,1:RETURN
950 CURSET 30,135,3:DRAW 20,-20,1:RETURN
960 CURSET 70,115,3:DRAW 0,20,1:RETURN
970 CURSET 70,140,3:CIRCLE 5,1:RETURN
980 CURSET 70,145,3:DRAW 0,25,1:CURSET 60,150,3:DRAW 20,0,1:RETURN
983 :
984 BC=0                    :REM pieds(optionnel)
985 CURSET 70,170,3:DRAW -5,15,1:CURSET 70,170,3:DRAW 5,15,1
986 CURSET 65,185,3:DRAW -3,-1,1:CURSET 75,185,3:DRAW 3,1,1:WAIT 10
987 CURSET 65,185,3:DRAW -3,-1,2:CURSET 75,185,3:DRAW 3,1,2
988 CURSET 65,185,3:DRAW -3,-3,1:CURSET 75,185,3:DRAW 3,3,1:WAIT 5
989 CURSET 65,185,3:DRAW -3,-3,2:CURSET 75,185,3:DRAW 3,3,2
990 BC=BC+1:IF BC<20  THEN 985
992 RETURN
1000 DATA  MAISON,SAPIN,BROUETTE,CIDRE,CASSETTE,VELO,PATINS,NOISETT
E,ORIC,*
```

Météorites

```
10 REM    METEORITES
20 :
22 REM    Une soucoupe se deplace a travers
23 REM    des meteorites.
24 REM    Utiliser les fleches pour se deplacer
30 :
40 SC$="@"
41 REM---------------- Definition soucoupe(optionnel)
42 DATA 12,12,30,63,51,51,63,63,63
45 A=46080:D=ASC(SC$)*8
46 FOR N=0 TO 7:READ ND:POKE A+D+N,ND:NEXT N
48 REM-----
50 PRINT CHR$(17);             :REM suppression curseur
55 :
60 XV=15:YV=2:AX=XV:AY=YV
90 CLS:PAPER6:INK4
95 PRINT:PRINT:PRINT:PRINT
100 FOR L=1 TO 10000
150 :X=RND(1)*35+1:PRINT SPC(X);"X"
160 :
175 :IF SCRN(XV,YV)<>ASC("X") THEN 180:REM test collision
176 :
178 :EXPLODE:PAPER5:WAIT 30:PAPER7:PLOT 20,23,"SCORE:"+STR$(L)
179 :WAIT 500:GOTO 60
180 :
181 :PLOT AX,AY-1," "      :REM effacement ancienne position
182 :PLOT XV,YV,SC$        :REM nouvelle position
185 :
186 :AX=XV:AY=YV           :REM ancienne position
187 :
188 REM------- test clavier
190 :X=PEEK(#208):IF X=56 THEN 300
200 :IF X=188  THEN IF XV<36 THEN XV=XV+1
210 :IF X=172  THEN IF XV>2 THEN XV=XV-1
300 NEXT L
310 REM----------------------------------------
320 REM   PEEK(#208) peut etre remplace par X$=KEY$:IF x$="" THEN

330 :
340 REM  pour obtenir une soucoupe rouge,faire:
350 REM  182 PLOT XV-1,YV,CHR$(1)+SC$+CHR$(4)
```



Bombardement ville

```
10 REM      BOMBARDEMENT VILLE
20 :
25 REM    Une soucoupe bombarde des immeubles
26 REM
45 PRINT CHR$(17)       :REM suppression curseur ctrl Q
50 REM-------------- redefinition graphique de '!'
60 DATA 63,63,51,51,51,63,63,63
70 A=46080:D=ASC("!")*8
80 FOR N=0 TO 7:READ ND:POKE A+D+N,ND:NEXT N
81 REM----------- Definition soucoupe
82 DATA 12,12,30,63,63,63,63,0
83 SC$="@":D=ASC(SC$)*8
84 FOR N=0 TO 7:READ ND:POKE A+D+N,ND:NEXT N
85 REM------------- Definition projectile
86 DATA 0,0,0,0,0,0,0,6
87 P$="#":D=ASC(P$)*8
88 FOR N=0 TO 7:READ ND:POKE A+D+N,ND:NEXT N
90 REM--------- Dessin 5 immeubles
100 X0=1:Y0=23:XB=X0:YB=Y0
110 CLS:PAPER0:INK6::PLOT 1,0 ,"Appuyer sur <espace>"
120 FOR N=1 TO 5
130 :H=RND(1)*6+5:L=RND(1)*4+3
150 :C=RND(1)*7+1:GOSUB 200
170 :XB=XB+L+1
180 NEXT N
185 GOTO 370
190 REM--------- dessin 1 immeuble
200 FOR Y=YB TO YB-H STEP-1
210 :AF$=CHR$(C)
220 :FOR X=1 TO L:AF$=AF$+"!":NEXT X
230 :PLOT XB,Y,AF$
240 NEXT Y
250 RETURN
```

```
370 REM================== BOMBARDEMENT =======
400 PLAY 1,0,0,0
420 YS=2:X1=2:X2=37:S=1              :REM coordonnees soucoupe/sens
430 :
440 FOR D=X1 TO X2 STEP S
450 :XS=D
454 :IF YS>17 THEN WAIT 300:GOTO 100       :REM Fin?
455 :IF SCRN(XS,YS)=33 THEN EXPLODE:PAPER 5:WAIT 20:PAPER0:WAIT 400
:GOTO 100
460 :PLOT XS-S,YS,1:PLOT XS,YS,SC$    :REM soucoupe rouge
470 :
480 :IF KEY$="" THEN 500       :REM depart tir
490 :FOR I=1 TO 4
495 : IF YP(I)=0 THEN YP(I)=YS+1:XP(I)=XS:GOTO 500
497 :NEXT I
498 :
500 :FOR I=1 TO 4              :REM avance projectiles
510 : IF YP(I)=0 THEN 580
520 : IF SCRN(XP(I),YP(I))=32 THEN 550
530 :
550 : PLOT XP(I),YP(I)," "
560 : IF YP(I)<23 THEN YP(I)=YP(I)+1:PLOT XP(I),YP(I),P$
565 : IF YP(I)=23 THEN YP(I)=0:XP(I)=0
580 :NEXT I
590 :SOUND 1,D*3,12
600 NEXT D
610 PLOT XS,YS," ":X=X1:X1=X2:X2=X:S=-S      :REM  Inversion sens
620 YS=YS+1
630 GOTO 440
```

Tir sur cible numérique

```
10 REM   TIR SUR CIBLE NUMERIQUE
20 REM
30 REM   Une cible de 5 chiffres defile
35 REM   Le score est proportionnel
37 REM   au chiffre frappe et inversement
40 REM   proportionnel au temps de reflexion
45 CLS:PLOT 18,14,CHR$(1)+"Appuyer sur <ESPACE>"
50 :
55 CB$="":DP=0:N=0
58 REM -------------- Choix de chiffres au hasard ------
60 FOR I=1 TO 5
70 :CB$=CB$+STR$(INT(RND(1)*9)+1)
80 NEXT I
90 REM-----
95 YT=14:XT=15           :REM coordonnees
96 XC=10:YC=1            :REM cible
100 PLOT XC,YC,CB$       :REM affichage cible
102 WAIT 200
103 TP=250               :REM temps partie
105 REM-------------------
110 PLOT XC,YC,CB$
115 :
120 CB$=RIGHT$(CB$,2)+LEFT$(CB$,LEN(CB$)-2)
130 PLOT XT,YT,"X"
200 REM----
210 IF KEY$<>"" THEN DP=1     :REM Test depart
220 :
230 IF DP=0 THEN 300
235 :
240 PLOT XT,YT," "
255 YT=YT-1
260 PLOT XT,YT,"X"
265 IF YT=YC+1 THEN 500
300 :
310 N=N+1:IF N=20  THEN N=0:GOTO 110
315 TP=TP+1
320 GOTO 210
490 REM--------------- RESULTAT ----
500 :
505 P$=MID$(CB$,XT-XC+3,1)
510 PLOT 1,20,"POINTS:"+CHR$(2)+P$
512 PLOT 1,21,"TEMPS:  "+STR$(TP/100)
520 SC=SC+VAL(P$)
525 PLOT 15,20,"TOTAL:"+STR$(SC)
526 T=T+TP
527 PLOT 15,21,"TOTAL:"+STR$(T/100)
528 PLOT 15,23,"TOT/TEMPS:"+STR$(SC/T*100)
530 WAIT 300
535 PLOT XT,YT," "
540 GOTO 50
```

83579

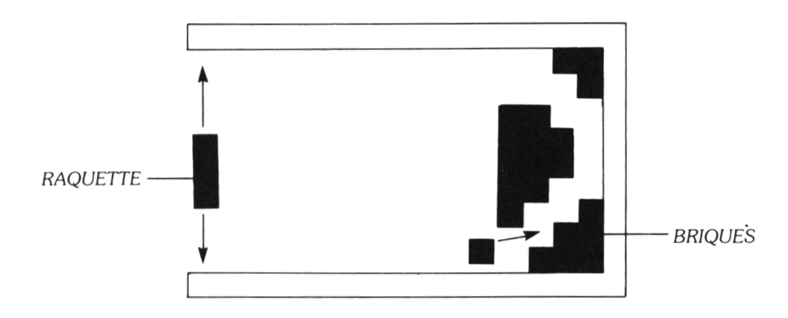
X

POINTS : 100
TEMPS : 47

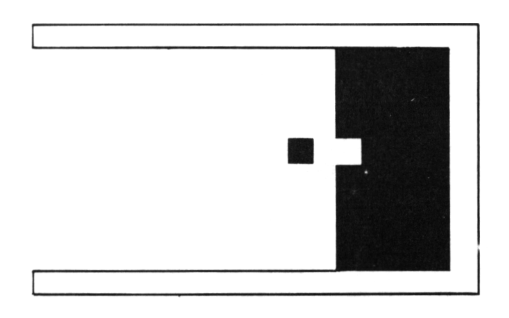
CASSE BRIQUES

- Une balle se déplace entre trois murs. Elle détruit des briques dans le mur de droite.
- Le joueur doit faire rebondir la balle avec une raquette qu'il contrôle avec les flèches ↑↓. Si le joueur rate la balle, le score est affiché (nombres de briques détruites.)

Version 1 :
La balle détruit toutes les briques sur son passage.



Version 2 :
La balle ne détruit qu'une brique puis rebondit.



Casse briques n° 1

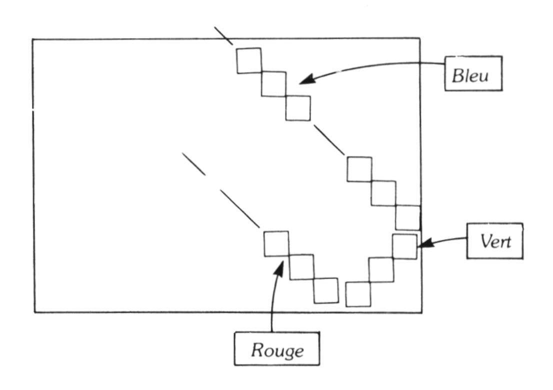
```
10 REM    CASSE BRIQUES NO1
15 PRINT CHR$(6);
20 :
25 LORES 0
30 X1=1:X2=35:Y1=2:Y2=23
35 X3=X2-5
40 REM--------------------------- Dessin terrain ----------
50 FOR X=X1 TO X2:PLOT X,Y1,17:PLOT X,Y2,17:NEXT X
60 FOR Y=Y1 TO Y2:PLOT X2,Y,17:NEXT Y
65 FOR Y=Y1+1 TO Y2-1
66 FOR X=X3 TO X2-1:PLOT X,Y,19:NEXT X
67 NEXT Y
70 REM---------------------- Initialisations -------------
80 VX=1 :VY=1
90 XV=VX:YV=VY
100 XB=5:YB=INT(RND(1)*10)+5
110 PLOT XB,YB,18
120 XR=2:YR=16                 :REM Raquette
130 PLOT XR,YR,17:PLOT XR,YR+1,17:PLOT XR,YR+2,17
140 REM ---------------------- Avance balle -------------
150 :
160 AX=XB:AY=YB                :REM Ancienne Position
170 :
180 XB=XB+XV:YB=YB+YV          :REM Nouvelle Position
185 :
190 IF XB=15 THEN XB=XB+XV*INT(RND(1)*2)
200 :
210 IF XB=>X2 THEN XV=-VX:YB=YB-YV:GOTO 180
220 IF YB<Y1+1 THEN YV=VY:GOTO 180 :REM Reb cotes
230 IF YB=>Y2 THEN YV=-VY:GOTO 180
240 :
245 :                                   :REM Reb raquette
250 IF XB<XR+1 THEN IF YB=>YR THEN IF YB<YR+3 THEN XV=VX:GOTO 180
255 :
257 IF SCRN(AX,AY)=19 THEN SC=SC+1
260 PLOT AX,AY,16                       :REM Eff balle
268 IF SCRN(XB,YB)=19 THEN SC=SC+1
269 :
270 PLOT XB,YB,18
272 IF SCRN(XB+1 ,YB)=19 THEN PLOT XB+1,YB,16:SC=SC+1
275 :
280 IF XB<=XR THEN PLOT 20,10,"SCORE:"+STR$(SC):WAIT 300:SC=0:GOTO
20
290 REM --------------- Deplacement raquette ------------
300 X=PEEK(#208):IF X=56 THEN 330
305 :
310 IFX=180 THEN IF YR<Y2-3 THEN YR=YR+1:PLOT XR,YR-1,16:PLOT XR,YR
+2,17
320 IFX=156 THEN IF YR>Y1+1 THEN YR=YR-1:PLOT XR,YR+3,16:PLOT XR,YR
,17
330 GOTO 160
```

Casse briques n° 2

```
10 REM  CASSE BRIQUES NO2
15 PRINT CHR$(6);
20 :
25 LORES 0
30 X1=1:X2=35:Y1=2:Y2=23
35 X3=X2-5
40 REM--------------------- Dessin terrain ---------
50 FOR X=X1 TO X2:PLOT X,Y1,17:PLOT X,Y2,17:NEXT X
60 FOR Y=Y1 TO Y2:PLOT X2,Y,17:NEXT Y
65 FOR Y=Y1+1 TO Y2-1
66 FOR X=X3 TO X2-1:PLOT X,Y,19:NEXT X
67 NEXT Y
70 REM----------------- Initialisations ------------
80 VX=1 :VY=1
90 XV=VX:YV=VY
100 XB=5:YB=INT(RND(1)*10)+5
110 PLOT XB,YB,18
120 XR=2:YR=16               :REM Raquette
130 PLOT XR,YR,17:PLOT XR,YR+1,17:PLOT XR,YR+2,17
140 REM ------------------- Avance balle ------------
150 :
160 AX=XB:AY=YB              :REM Ancienne position
170 :
180 XB=XB+XV:YB=YB+YV        :REM Nouvelle position
185 :
190 :
200 :
210 IF XB=>X2 THEN XV=-VX:YB=YB-YV:GOTO 180
220 IF YB<Y1+1 THEN YV=VY:GOTO 180 :REM Reb cotes
230 IF YB=>Y2 THEN YV=-VY:GOTO 180
240 :
245 :                                    :REM Reb raquette
250 IF XB<XR+1 THEN IF YB=>YR THEN IF YB<YR+3 THEN XV=VX:GOTO 180
255 :
257 IF SCRN(AX,AY)=19 THEN SC=SC+1
260 PLOT AX,AY,16                :REM Eff balle
268 IF SCRN(XB,YB)=19 THEN SC=SC+1
269 :
270 PLOT XB,YB,18
272 IFSCRN(XB+1,YB)=19THEN PLOTXB+1,YB,16:XV=-VX:PLOTXB,YB,16:SC=SC
+1:GOTO 180
275 :
280 IF XB<=XR THEN PLOT 20,10,"SCORE:"+STR$(SC):WAIT 300:SC=0:GOTO
20
290 REM ------------- Deplacement raquette ------------
300 X=PEEK(#208):IF X=56 THEN 330
305 :
310 IFX=180 THEN IF YR<Y2-3 THEN YR=YR+1:PLOT XR,YR-1,16:PLOT XR,YR
+2,17
320 IFX=156 THEN IF YR>Y1+1 THEN YR=YR-1:PLOT XR,YR+3,16:PLOT XR,YR
,17
330 GOTO 160
```

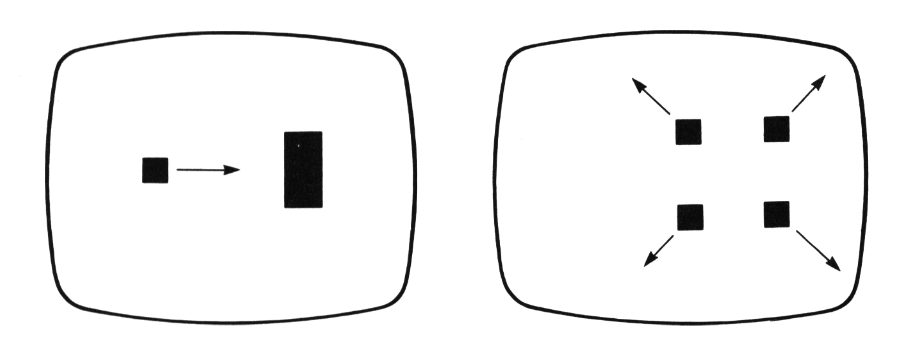
Tapisserie

```
10 REM         TAPISSERIE
20 :
22 REM    Un point se deplace sur l'ecran
23 REM    en laissant une trace de couleur
24 REM    la couleur change a chaque rebond
25 REM
30 LORES0
32 XB=2:XH=38              :REM bornes x
34 YB=2:YH=23              :REM bornes y
40 :
90 :
130 :
140 XP=XB+2:YP=YB+3        :REM coordonnees point mobile
150 VX=1:VY=1
160 :
170 IF XP<XB+1 OR XP>XH-1 THEN VX=-VX: C=INT(RND(1)*7)+1:GOTO 220
180 IF YP<YB+1 OR YP>YH-1 THEN VY=-VY:C=INT(RND(1)*7)+1:GOTO 220
190 :
200 PLOT XP,YP,16+C
205 :
220 XP=XP+VX:YP=YP+VY
230 GOTO 170
```



Explosion

```
10 REM    EXPLOSION
12 :
13 REM    Un point se deplace vers une cible.
14 REM    Au moment ou il atteint la cible,
15 REM    celle ci se disperse dans 4 directions.
20 :
22 REM
30 TIR$=CHR$(17)
35 :
40 LORES0
50 Y=15
52 REM--------- cible
55 XC=30:YC=Y:PLOT XC,YC,20:PLOT XC,YC-1,20
57 REM--------------deplacement tir
60 FOR X=1 TO 38
65 :PLOT X-1,Y ,16
70 :PLOT X,Y ,TIR$
75 :IF X=XC THEN PLOTXC,YC,16:PLOT XC,YC-1,16:EXPLODE:GOTO 190
80 NEXT X
90 END
180 REM---------------------- explosion
190 FOR N=1 TO 5  STEP 2
210 : PLOT X+N,Y+N,18:PLOT X+N-2,Y+N-2,16
220 : PLOT X+N,Y-N,17:PLOT X+N-2,Y-N+2,16
230 :PLOT X-N,Y+N,19:PLOT X-N+2,Y+N-2,16
240 : PLOT X-N,Y-N,20:PLOT X-N+2,Y-N+2,16
300 NEXT N
310 N=N-2
320 PLOT X+N,Y+N,16:PLOT X+N,Y-N,16
330 PLOT X-N,Y+N,16:PLOT X-N,Y-N,16
340 :
350 WAIT 100:GOTO 40
```



Sapins

```
10 REM        SAPINS
12 :
15 REM  Des sapins de taille aleatoire
16 REM  remplissent l'ecran.
20 :
45 HIRES:INK2:PAPER7
80 :
290 REM-----------------
300 FOR NS=1 TO 15           :REM 15 sapins
310 :X0=20+RND(1)*180:Y0=1+RND(1)*130
320 :H=7+RND(1)*10
330 :GOSUB 480
340 NEXT NS
350 WAIT 300:END
480 REM------------ SAPIN
490 FOR N=1 TO 4
500 :FOR I=1 TO H
510 :CURSET X0-I/2,Y0+I,1
520 :DRAW I,0,1
530 NEXT I
540 :
545 :Y0=Y0+H/2:H=H*1.2
550 NEXT N
555 CURSET X0,Y0,1:DRAW 0,H,1
560 RETURN
```

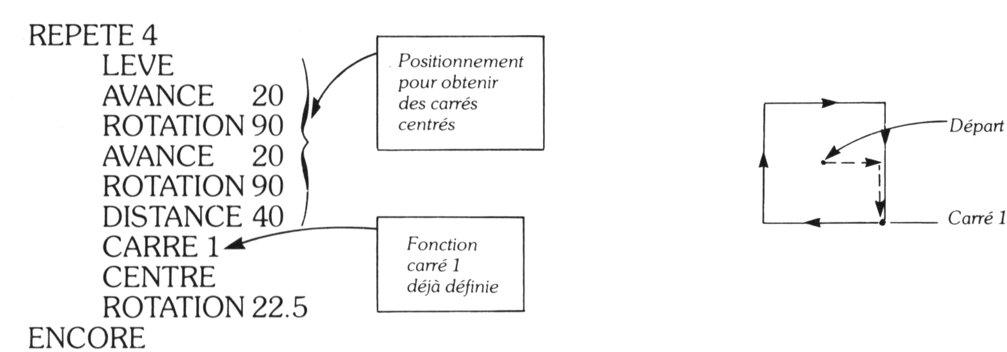
SIMULATEUR LOGO

- Nous reprenons le simulateur logo présenté dans 'ORIC 1 POUR TOUS'.
  Nous lui avons ajouté quelques fonctions.

DIS/ X : Division de la distance courante par X.
DI✳ X : Multiplication de la distance courante par X.
CENTRE : Positionne le point de départ.
ENCRE X : La couleur d'écriture devient celle spécifiée (1-7)

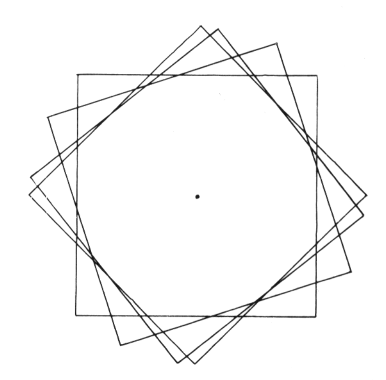
Exemple N° 1 :
Tracé des carrés centrés avec rotation.

```
REPETE 4
      LEVE
      AVANCE    20
      ROTATION 90
      AVANCE    20
      ROTATION 90
      DISTANCE 40
      CARRE 1
      CENTRE
      ROTATION 22.5
ENCORE
```



- Afin d'obtenir des carrés centrés, nous devons positionner les carrés avec :

```
AVANCE    20
ROTATION 90
AVANCE    20
ROTATION 90
```



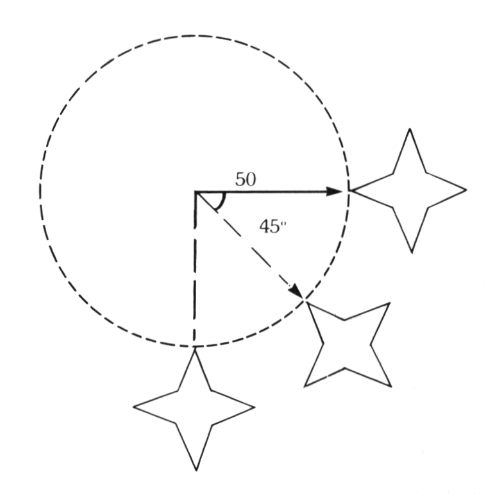
- Pour revenir au centre, nous avons créé une fonction 'centre'. Celle-ci ne modifie pas l'angle en cours.

Exemple N° 2 :

Tracé d'étoiles sur un cercle

```
REPETE 8
      LEVE
      ROTATION    45
      AVANCE      50
      DISTANCE    15
      ETOILE
      LEVE
      AVANCE     – 50
ENCORE
```

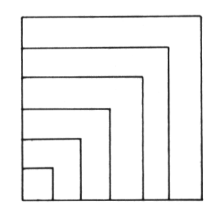
Fonction déjà définie



Exemple N° 3 :

Carrés emboîtés

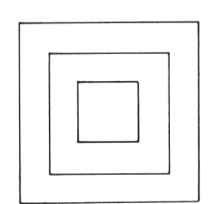
```
DISTANCE 30
REPETE 4
      CARRE       1
      DIS/        1.2
ENCORE
```



Exemple N° 4 :

Carrés centrés

```
DISTANCE 30
REPETE 4
      LEVE
      DIS/        2
      AVANCE
      ROTATION 90
      AVANCE
      ROTATION 90
      DIS*        2
      CARRE       1
      CENTRE
      ROTATION 180
      DIS*        1.2
ENCORE
```



Entrée des commandes avec INPUT :

- L'instruction 550 READX$ peut être remplacée par : INPUT 'commande ?'; X$
- Les commandes ne sont exécutées qu'après la frappe de '✳'.

Exemple :

Commande ? AVANCE 20 ↵
ROTATION 90 ↵
AVANCE 30 ↵
✳ ↵

AMELIORATION POSSIBLE :

Nous pourrions définir des variables :

X = 30
AVANCE X/2

Il faudrait pour cela créer une table des variables et de leurs valeurs.

VR$( )	VL( )
X	30
Y	20
Z	50

Logobis

```
10 REM LOGOBIS SIMULATEUR LOGO ---------------
20 :
30 HIRES:INK3
35 HIMEM #97FF
40 :
50 DIM CD$(100)
55 DIM F$(20,20)
60 X0=100:Y0=100:CX=1:CY=0:DIST=20
70 REM ---- Pseudo Programme interprete -----
80 :
90 REM------------------ definition etoile
100 DATA DEFI ETOILE
110 DATA BAISSE
115 DATA REPETE 4
120 DATA  AVANCE
125 DATA  ROTATION 160
130 DATA  AVANCE
135 DATA  ROTATION -70
140 DATA ENCORE
145 DATA FINI
150 REM---------------- definition carre
160 DATA DEFI CARRE1
165 DATA BAISSE
170 DATA REPETE 4
180 DATA  AVANCE
190 DATA  ROTATION 90
200 DATA ENCORE
210 DATA FINI
220 REM------------------- trace carres centres avec rotation
235 DATA ENCRE 5
240 DATA REPETE 4
241 DATA  LEVE
242 DATA  AVANCE 20
243 DATA  ROTATION 90
244 DATA  AVANCE 20
245 DATA  ROTATION 90
246 DATA  DISTANCE 40
247 DATA  CARRE1
250 DATA  CENTRE
252 DATA  ROTATION 180
253 DATA  ROTATION 22.5
260 DATA ENCORE
300 :
310 :
311 DATA CENTRE
312 REM------------------- trace etoiles sur un cercle
320 DATA REPETE 8
325 DATA LEVE
326 DATA  ROTATION 45
327 DATA  AVANCE 50
328 DATA  DISTANCE 15
330 DATA  ETOILE
335 DATA  LEVE
340 DATA  AVANCE -50
350 DATA ENCORE
```

```
351 :
352 :
353 :
354 :
355 :
356 :
357 :
358 :
359 :
360 REM------------------ carres emboites
370 DATA LEVE
375 DATA ROTATION -90
380 DATA AVANCE 70
400 DATA DISTANCE 30
402 DATA REPETE 4
404 DATA  CARRE1
406 DATA  DIS/ 1.2
410 DATA ENCORE
420 REM----------------- carres concentriques
425 DATA CENTRE
430 DATA DISTANCE 15
440 DATA REPETE 3
445 DATA  LEVE
450 DATA  DIS/ 2
460 DATA  AVANCE
470 DATA  ROTATION 90
495 DATA  AVANCE
498 DATA  ROTATION 90
500 DATA  DIS* 2
502 DATA  CARRE1
504 DATA  CENTRE
506 DATA  ROTATION 180
507 DATA  DIS* 1.1
510 DATA ENCORE
518 DATA *
520 REM ----------- Lecture des commandes dans CD$() --
530 I=1
540 :
550 READ X$ : PRINT X$
560 :
570 IF LEFT$(X$,4)="DEFI" THEN GOSUB 1250 : GOTO 550
580 :
590 IF X$="*" THEN NC=I : GOTO 670
600 FOR K=1 TO 5
610 IF X$=F$(K,0) THEN GOSUB 1330
620 NEXT K
630 :
640 CD$(I)=X$
650 I=I+1 : GOTO 550
651 :
652 :
653 :
654 :
655 :
656 :
657 :
```

```
658 :
659 :
660 REM-----------
670 PL=1 : REM -- Pointeur de ligne courante --
680 :
690 LG$=CD$(PL)
700 C$=LEFT$(LG$,4)
710 IF C$="REPE" THEN GOTO 930
720 IF C$="ENCO" THEN GOTO 970
730 IF C$="AVAN" THEN GOSUB 1080
740 IF C$="ROTA" THEN GOSUB 1000
750 IF C$="DIST" THEN GOSUB 860:DIST=VAL(RIGHT$(LG$,LEN(LG$)-P))
760 IF C$="DIS+"  THEN GOSUB 1210
765 IF C$="DIS/" THEN GOSUB 1500
770 IF C$="LEVE" THEN LV=1
780 IF C$="BAIS" THEN LV=0
795 IF C$="CENT"  THEN X0=100:Y0=100
796 IF C$="ENCR" THEN GOSUB 1540
797 IF C$="DIS*" THEN GOSUB 1600
799 PL=PL+1:IF PL>NC THEN END
800 GOTO 690
845 :
850 REM ---------- -------------- recherche espace -----
860 FOR P=1 TO LEN(LG$)
870 IF MID$(LG$,P,1)=" " THEN RETURN
880 NEXT P
890 P=0 : RETURN
900 REM ------------------------------- repete ---------
910 REM       PR :pt pile RP():contient no ligne apres REPETE
920 REM       NB() contenant nb de boucles
930 GOSUB 860
940 PR=PR+1:RP(PR)=PL+1:NB(PR)=VAL(RIGHT$(LG$,LEN(LG$)-P))
950 PL=PL+1 : GOTO 690
960 REM ----------------- encore -------.
970 NB(PR)=NB(PR)-1:IF NB(PR)>0 THEN PL=RP(PR):GOTO 690
980 PR=PR-1:PL=PL+1:GOTO 690
990 REM ---------------------------------- Angle -----------
1000 GOSUB 860
1010 ANG=ANG+VAL(RIGHT$(LG$,LEN(LG$)-P))
1020 IF ANG=>360 THEN ANG=ANG-360
1030 AR=ANG/360*6.28318
1040 CX=COS(AR):CY=SIN(AR)
1045 IF CX>-.01 AND CX<.01 THEN CX=0
1050 RETURN
1051 :
1052 :
1053 :
1054 :
1055 :
1056 :
1057 :
```

```
1060 REM ------------------------------- Trace droite ----
1070 :
1080 GOSUB 860
1090 IF P=0 THEN 1110
1095 V=VAL(RIGHT$(LG$,LEN(LG$)-P)):IF V=0 THEN 1110
1100 DIST=VAL(RIGHT$(LG$,LEN(LG$)-P))
1110 DX=DIST*CX:DY=DIST*CY*1.3
1112 IF DX=0 AND DY=0 THEN PRINT "ERR:";LG$:WAIT 200:RETURN
1115 IF LV=1 THEN X0=X0+DX:Y0=Y0+DY:RETURN
1116 :
1118 IF X0>0 THEN IF X0<238 THEN IF Y0>0 THEN IF Y0<198 THEN 1120
1119 PRINT "ERR:";X0,Y0,LG$:WAIT 300:RETURN
1120 CURSET X0,Y0,1
1125 X0=X0+DX:Y0=Y0+DY
1126 IF X0>0 THEN IF X0<238 THEN IF Y0>0 THEN IF Y0<198 THEN 1130
1127 PRINT "ERR:";X0,Y0,LG$:WAIT 300:RETURN
1130 DRAW DX,DY,1
1140 :
1190 RETURN
1200 REM ------------------- Increment distance ---
1210 GOSUB 860
1220 DIST=DIST+VAL(RIGHT$(LG$,LEN(LG$)-P))
1230 RETURN
1240 REM ------------------- Stockage fonction dans f$() ----
1250 NF=NF+1
1260 F$(NF,0)=RIGHT$(X$,LEN(X$)-5)
1270 FOR P=1 TO 20
1280 READ F$(NF,P)
1290 IF F$(NF,P)="FINI" THEN RETURN
1300 NEXT P
1310 STOP
1320 REM ------------------ Insertion fonction ----
1330 P=1
1340 :
1350 IF F$(K,P)="FINI" THEN RETURN
1360 CD$(I)=F$(K,P)
1370 P=P+1:I=I+1:GOTO 1350
1490 REM------------------ Division
1500 GOSUB 860:IF P=0 THEN RETURN
1505 V=VAL(RIGHT$(LG$,LEN(LG$)-P))
1510 IF V<>0 THEN DIST=DIST/V
1520 RETURN
1530 REM----------------- ENCRE
1540 GOSUB 860
1550 V=VAL(RIGHT$(LG$,LEN(LG$)-P))
1560 IF V<>0 AND C<8 THEN INK V
1570 RETURN
1590 REM------------------- Multiplication
1600 GOSUB 860:IF P=0 THEN RETURN
1610 V=VAL(RIGHT$(LG$,LEN(LG$)-P))
1620 IF V<>0 THEN DIST=DIST*V
1630 RETURN
```

Achevé d'imprimer en décembre 1983
sur les presses de l'imprimerie Laballery et Cie
58500 Clamecy
Dépôt légal : décembre 1983

N° d'impression : 312007
N° d'édition : 86595-120-1
ISBN : 2-86595-120-X

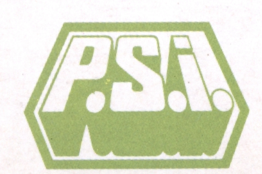


"Oric-ophiles" réjouissez-vous, voici **"52 programmes"** (nouvelle édition de l'ouvrage "30 programmes").

Faisant suite à l'ouvrage "Oric-1 pour tous" ce recueil s'articule sur quatre thèmes :

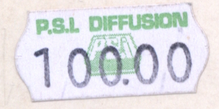
- EXERCICES, utilisant les fonctions particulières de l'Oric-1 telles que KEY, PLOT, FILL.
- ÉDUCATION, proposant des idées de programmes de géographie, orthographe, anglais, dessins...
- GESTION, développant quelques exemples du type annuaire téléphonique, fichiers d'adresses, courrier paramétré...
- JEUX, tous les grands classiques, Allumettes, Le compte est bon, Biorythmes, Jeu de la vie, Marienbad, et beaucoup d'autres, y sont traités avec originalité.

Tous ces programmes, largement commentés et illustrés vous permettront d'approfondir vos connaissances en Basic, et vous serviront de tremplin pour de nouvelles créations.



EDITIONS DU P.S.I.

B.P. 86 - 77402 LAGNY/MARNE CEDEX
FRANCE



***

52 PROGRAMMES ORIC-1 pour tous

JACQUES BOISGONTIER

P.S.I.